

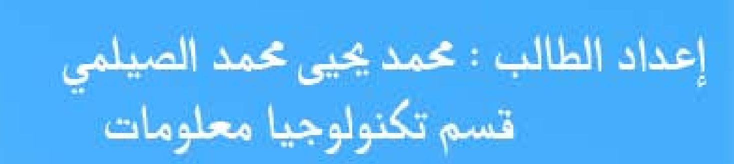
MYS-ME

2010

لغة جافا بإستخدام المحرر NetBeans 5.5









كتاب لتعليم الأساسيات في لغة (JAVA "1")

شرح // محمد يحيى محمد الصيلمي .

للمراسله ايميل :- me_mo_ry_123@hotmail.com

هذا الكتاب مجاني و لا اريد منكم سوى الدعاء لوالدي بالمغفرة و طول العمر ، و لكل المؤمنين ، و لي بالتوفيق في ديني و دنياي و رحلت دراستي مع الحاسوب .



آلسلآمـ، عـ.ـليكمـ، ،،،

المقدمة ،،

هذا الكتاب صممته لمساعدة من اراد تعلم اللغة ،
و قد قمت بتصميمه وفقا لما تعلمته من مهارات في أثناء دراستي
و أنا أقوم بكتابته قد راعيت المبتدئين بشكل كبير، و قد اهتممت بالشرح بالصورة
فهي ابلغ من الكتابة ، و قد تشرح الصورة ما تشرحه ألاف السطور ،
بالاضافة الى شرح النقاط بالتفصيل الملل لتعليم
المبتدأ الأساسيات
في اللغة ....

لغة جافا (1)

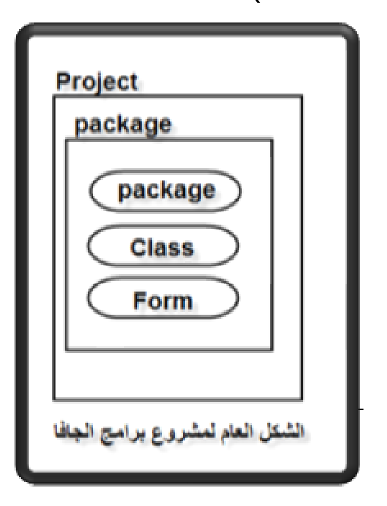
- هي لغة مفتوحة المصدر يتم تنزيلها من موقع sun java و شبيتها على الجهاز .
- و تحتاج برامجها الى مشغل وسيط في بيئة windows ويسمى JDK وهو ملف تحميلي خاص ببرامج الجافا فلا يعمل اي برنامج مصمم بلغة الجافا إلا بوجوده و تثبيته على النظام المشغل للبرنامج و يشبه عمله الى حد كبير عمل حزم framework.net الموجود في لغات الفيجول استديو (C# , vb.net , j#) .

و من مميزات هذي اللغة ان برامجها تعمل على جميع الانظمة لذلك تعتبر هذه اللغة من اقوى اللغات .

- و سيتم البرمجه في هذا الكتاب على احد محررات الجافا و الذي يسمى (Net beans) .

ملفات المشروع و الشكل العام لها :

- Project: وهي عبارة عن المشروع بشكل عام .
وقد يكون مشروع : web ، mobile ، Form .
- Package : وهي عبارة عن حزم لترتيب مجموعة الكلاسات (class) او مجموعة فورمات (Forms) .
- Class : هي شفرة برمجية تحتوي دوال و إجراءات ، بالإضافة الى المشيد .
- Form: و تحتوي على كلاسات بالاضافة الى الواجهات interface.

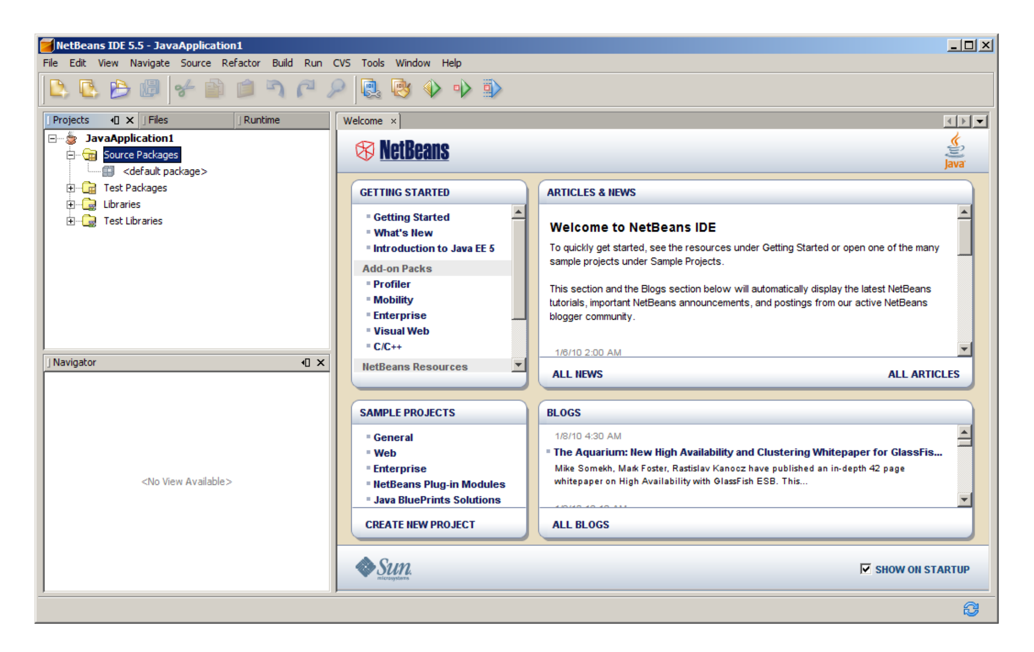


◄ طريقة فتح مشروع جديد :

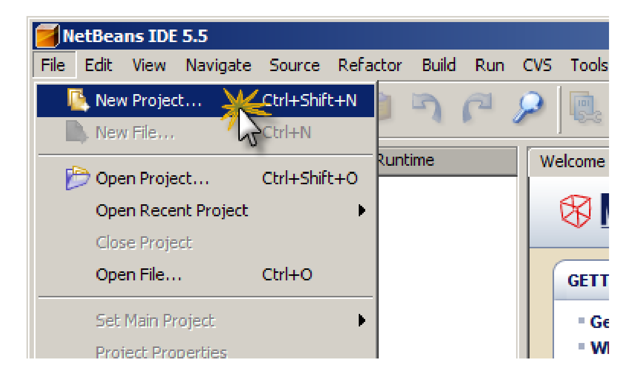
1. من على سطح المكتب نقوم بضغط مرتين على ايقونة محرر لغة جافا :



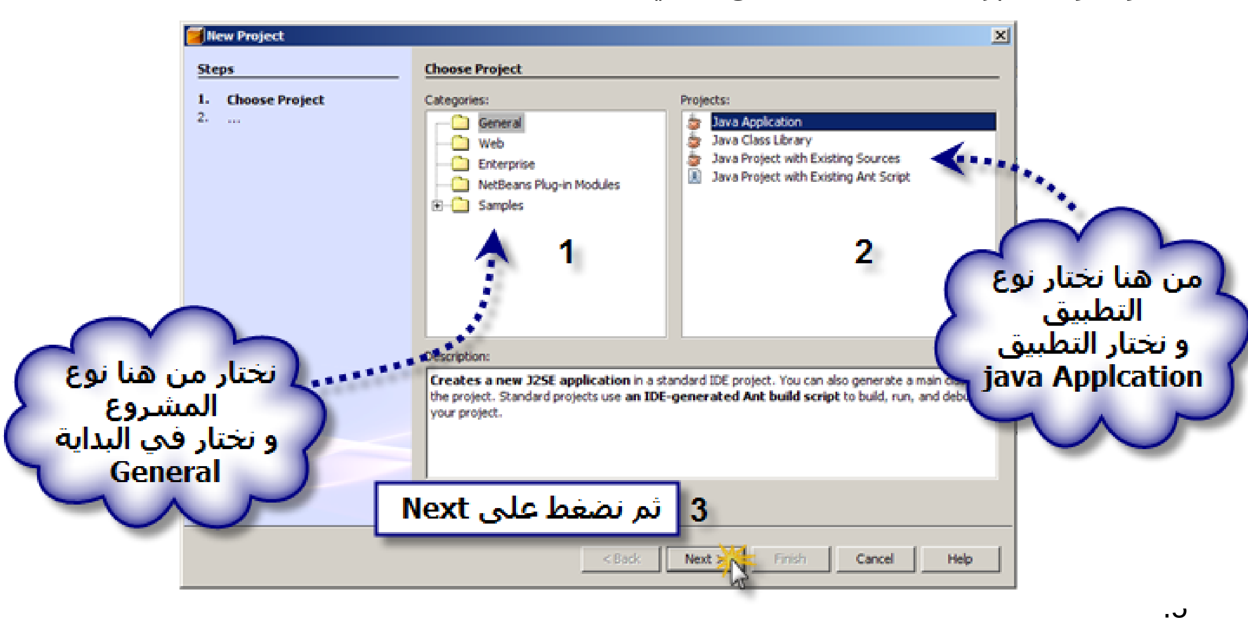
2. سوف تظهر النافذة الرئيسية للمحرر بالشكل التالي :



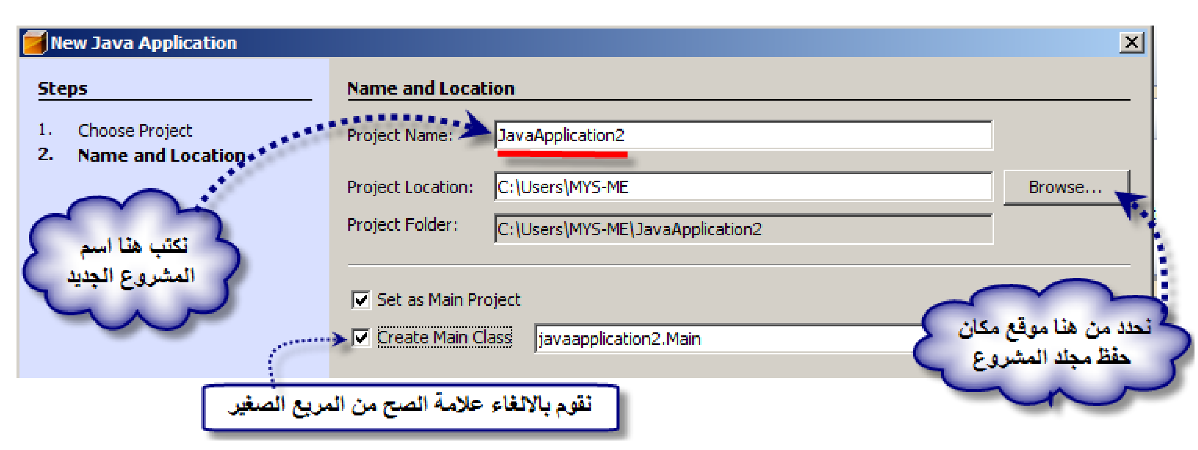
3. من قائمة File نختار الخيار New project .



4. و سوف تظهر نافذه جديده بالشكل التالي :



5.

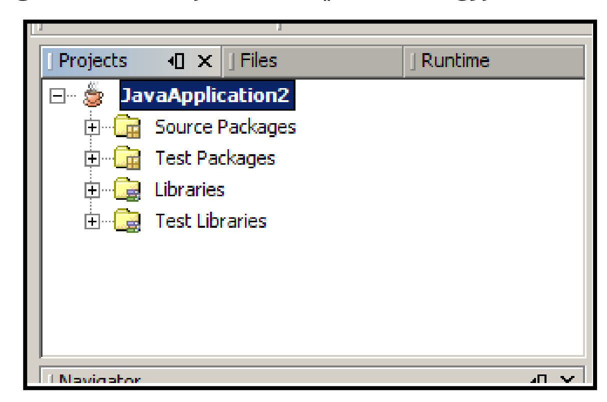


◄ شرح الصورة السابق :

- Set as main project : و يعني ذلك وضعه المشروع الاساسي و يمكن ربطه بباقي المشاريع .
- Create Main Class : وهو إجراء خاص يتم تنفيذه اول إجراء عند تشغيل البرنامج وهو بمثابة الحدث Loadفي vb.net، فعند تفعيله فإن البرنامج يقوم باجراء كلاس في البداية ، و الذي يقوم بعملية التحقق من المكاتب الموجوده في المجلد system32 و في حالة عدم وجودها فانه يقوم بنسخها ، و لكن حاليا يفضل عدم تفعيلها و الغاء الصح الذي امامها .

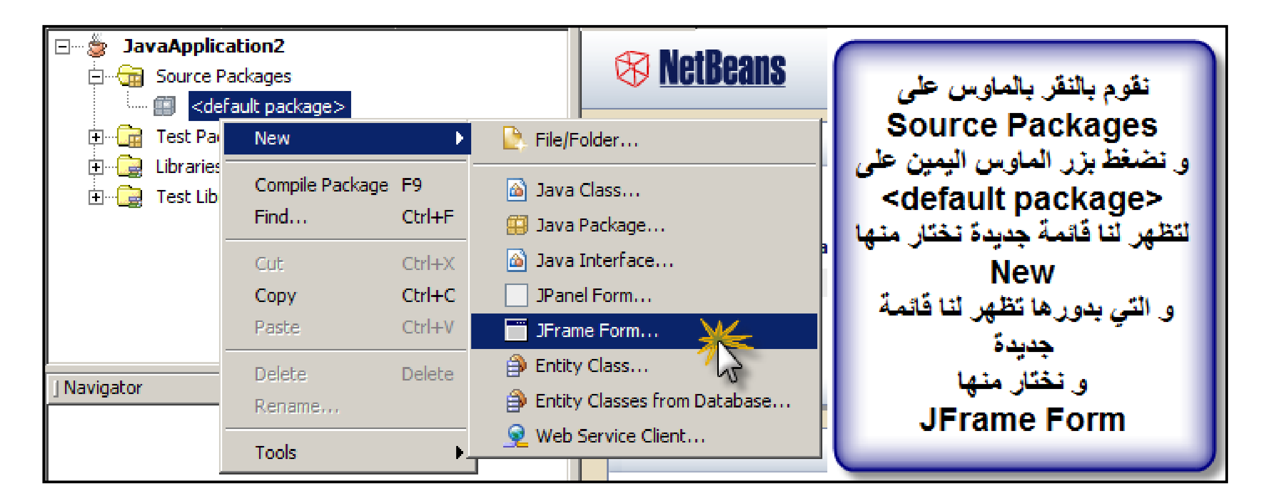
6. ثم تضغط على Finish . Finish

7. سوف تظهر ملفات المشروع الجديد في النافذة الرئيسيه بالشكل التالي :

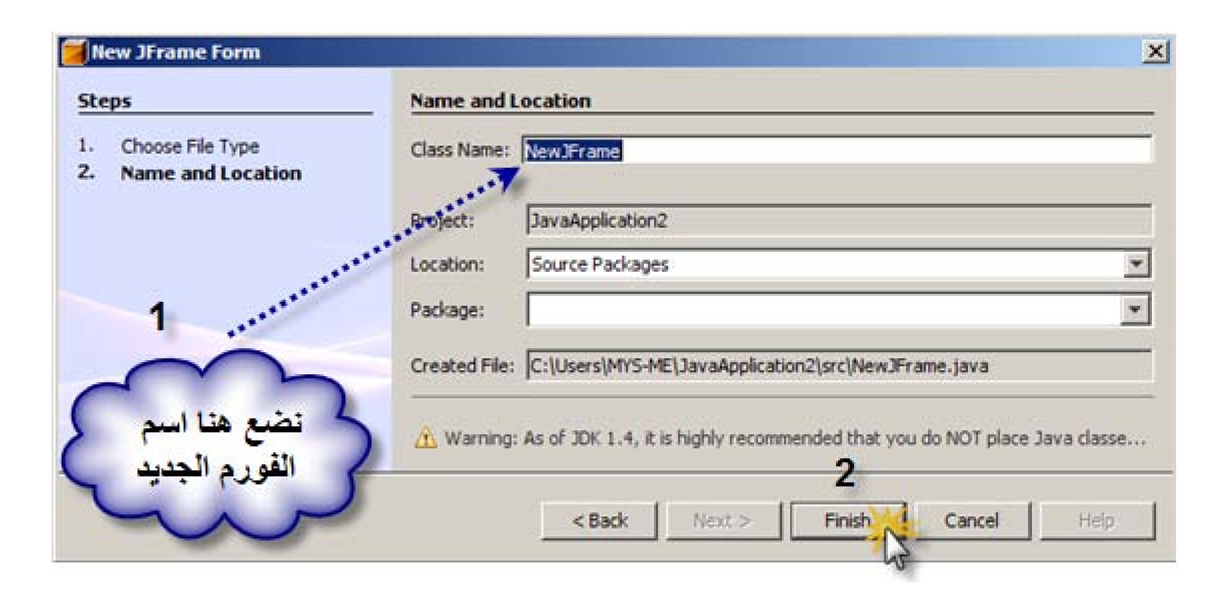


- حيث تمثل Libraries المكاتب التي يحتويها JDK .
- و تمثل Source Packages ملفات و نوافذ و كلاسات المشروع الذي نود تصميمه .

◄ ومن اجل ادراج فورم جديد لبدأ عملية تصميم المشروع ، نتبع التالي :



لتظهر لنا نافذه جديده بالشكل التالي :



وسوف يظهر **تبويب** جديد باسم الفورم الذي قمت بكتابته و ايضا اطار العمل عليه، بالشكل كالتالي :

نافذة الكائنات التي سوف تدرج الى الفورم

هنا واجهة التصميم و التي سيتم ادراج الكائنات فيها

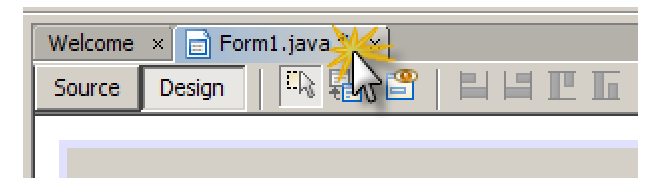
هنا الكائنات التي يتم ادراجها و لكن لا تظهر على الفورم مثل
timer , dailog

تحتوي هذه النافذه على ملفات و مكاتب البرنامج و تتيح للمستخدم التنقل بين الفورمات اذا تعددة

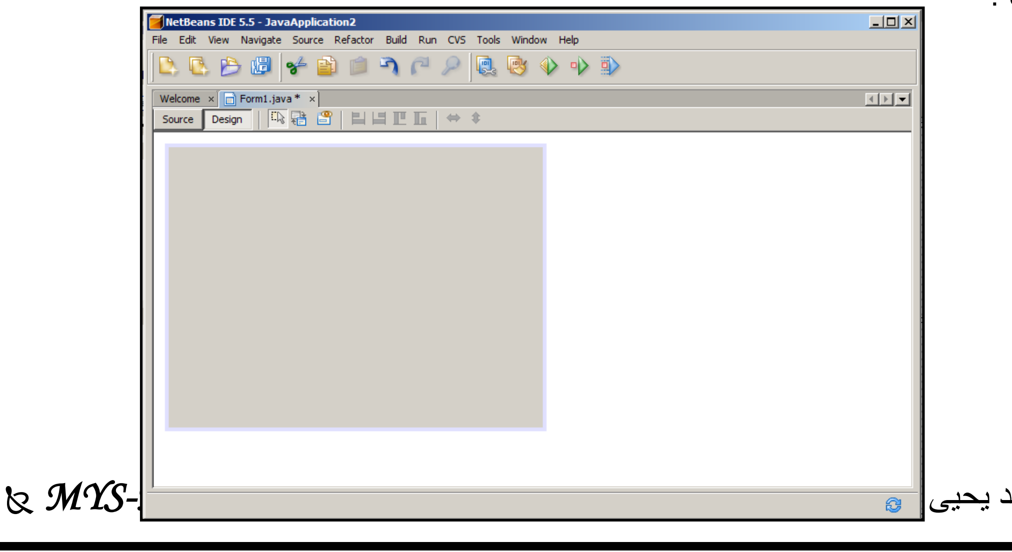
هنا نافذة خصائص الكائنات او الفورم

ملاحظات عامة :

1. من اجل تنفذ البرنامج الذي قمنا بتصميمه نضغط على الزر F5 من على الكيبورد ، او من خلال الضغط على الزر [زر التشغيل] الموجود اعلى البرنامج .
2. عند النقر المزدوج على تبويب نافذة الفورم كما في الشكل التالي :

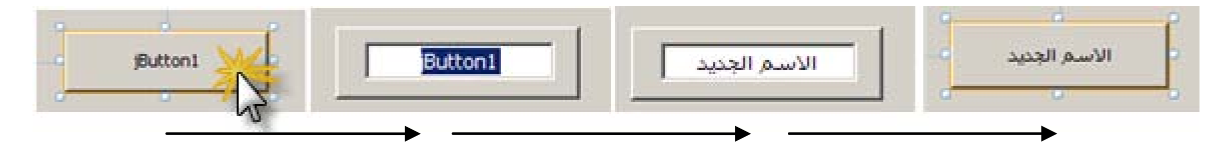


فإنه يتم تكبير نافذة الفورم و اظهارها بالكامل من اجل تحكم اكبر بشكل الفورم و الصورة التاليه توضح الفكره :

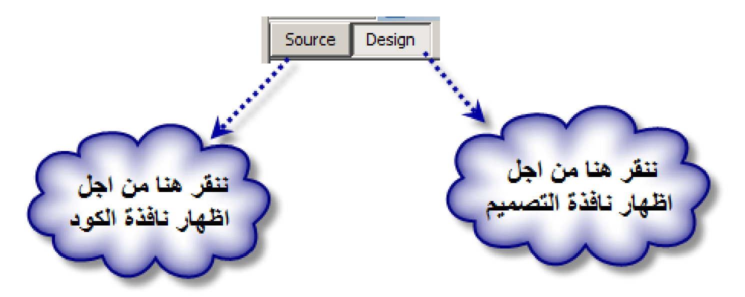


و من اجل الرجوع الى الوضعية السابقة يتم النقر المزدوج على التبويب نفسه .

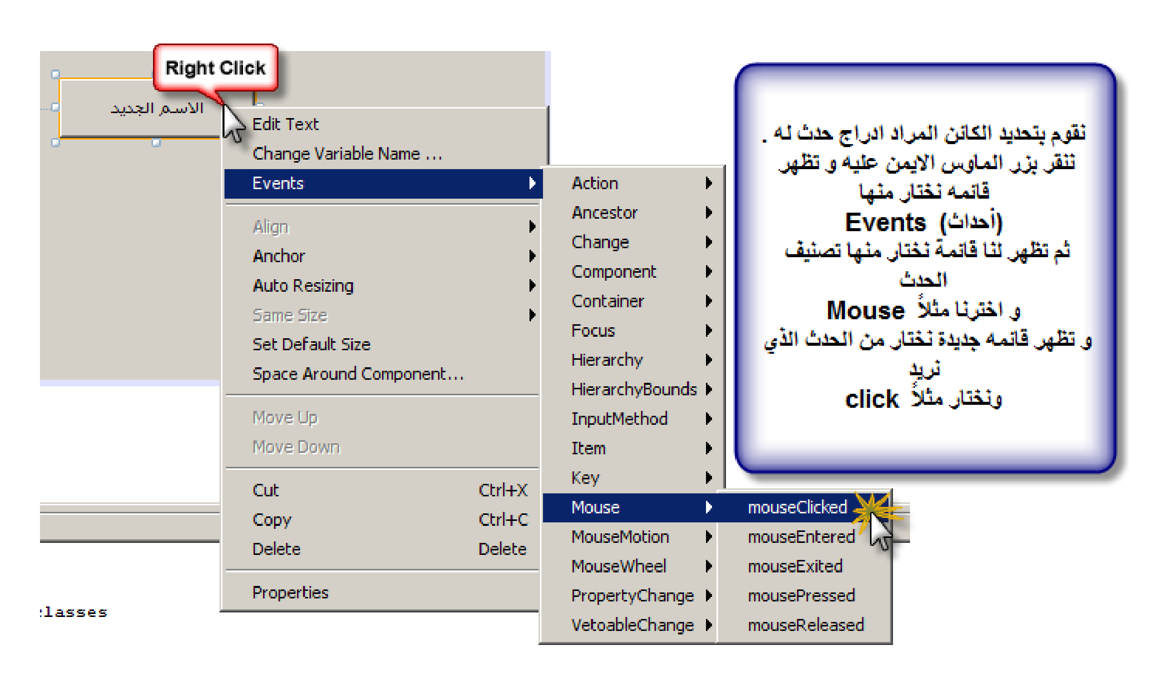
3. من اجل تغير الاسم الظاهري لاي كائن يتم النقر المزدوج عليه و تغير اسمه كالتالي :



4. من اجل التنقل بين نافذة الاكواد و نافذة التصميم (واجهة الفورم) نقوم بالتالي :



5. من اجل اختيار حدث لكائن معين و كتابة الكود بداخله نتبع التالي :



6. الشفرات الازمة لتشييد اي فورم (واجهة تصميم) :

عند فتح نافذة الكود نلاحظ بداية الكود بهذي **السطور** و التي تمثل عملية توريث و انشاء الفورم .

فـ extends :

تمثل التوريث للصفات من المكتبة javax.swing.Jframe الى الكائن Form1 .

و الـ javax.swing.JFrame : يمثل المكان الذي سيتم اخذ الصفات منه و توريثها للفورم .

و الـ initComponents :

يمثل عملية التشييد للفورم التي تم توريثه

في السطر السابق .

```
public class Form1 extends javax.swing.JFrame {

    /** Creates new form Form1 */
    public Form1() {
        initComponents();
    }
```

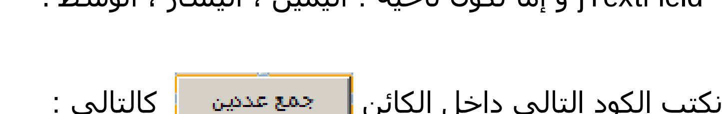
◄ **التطبيق الاول** : تصميم برنامج يقوم بجمع عددين ، و اظهار النتيجة .

سيكون شكل نافذة التطبيق كالتالي :

تعديل خصائص الكائن jTextField2 + jTextField1 :
كالتالي :- horizontal alignment = Center



وهي خاصية تتحكم في محاذات النص داخل الكائن
jTextField و إما تكون ناحية : اليمين ، اليسار ، الوسط .



نكتب الكود التالي داخل الكائن جمع عددين كالتالي :

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
double No1,No2,Res ;
No1 = Double.parseDouble(jTextField1.getText());
No2 = Double.parseDouble(jTextField2.getText());
Res = No1 + No2 ;
jLabel1.setText("" + Res);
    }
```

في السطر الاخير تم وضع علامة **اقتباس** في (" " + Res) ، و ذلك لجعل قيمة المتغير Res التي يرجعها نصيه ، و التي ستكون اساساً من النوع double و التي ستتحول تلقائياً من رقمية الى نصيه بسبب عملية دمجها بما تحتويه علامة الاقتباس و التي تحتوي فراغ بداخلها .

كما نلاحظ فقد تم كتابة الحرف الاول في كلمة **double** بالسطر الاول بالحالة الصغيره (**d**) ، بعكس **Double** الموجودة في السطر الثاني ، فما الفرق بينهما ؟
- **double** : حيث تمثل نوع بيانات (**Data type**) ، و يتم تعريف المتغيرات باستخدامها .
- **Double** : حيث تمثل استدعاء مكتبه (**Data class**) و استخدام دوالها مثل **parseDouble** .

شرح الكود السابق :

double No1,No2,Res ;

في السطر السابق تم تعريف ثلاثة متغيرات من النوع double

No1 = Double.parseDouble(jTextField1.getText());
No2 = Double.parseDouble(jTextField2.getText());

في السطرين السابقين تم اخذ النص الذي بداخل jTextField1 عن طريق الدالة getText و من ثما تحويل الارقام المدخله من نصيه الى رقميه عن طريق الدالة Double.parseDouble ومن ثما ادراج الرقم داخل المتغيرين No1,No2

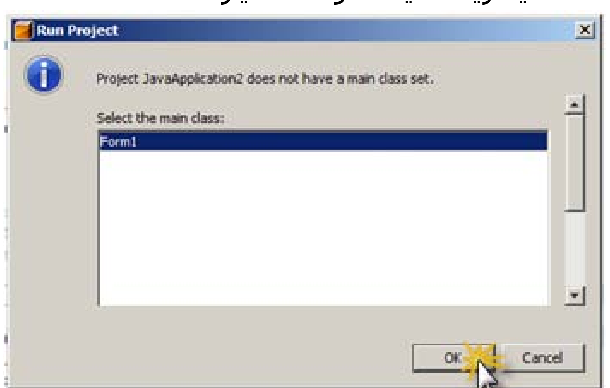
Res = No1 + No2 ;

عملية جمع المتغييرين و ادراج القيمة الجديدة في المتغير Res

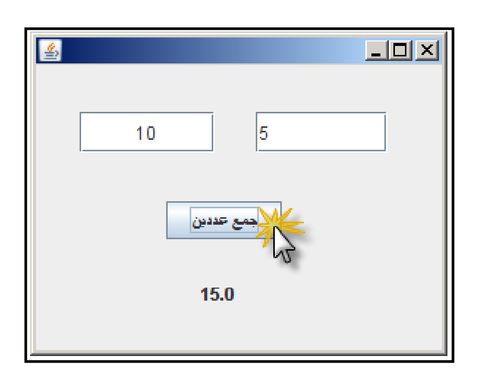
JLabel1.setText("" + Res);

يقوم بادراج القيمة الموجودة في المتغير Res الى الكائن JLabel1 عن الدالة

1. و عند تنفيذ البرنامج عن طريق الزر و ستظهر نافذه بالشكل التالي و نختار منها الفورم الذي نريد تنفيذه ، و قد اخترات هنا Form1 :

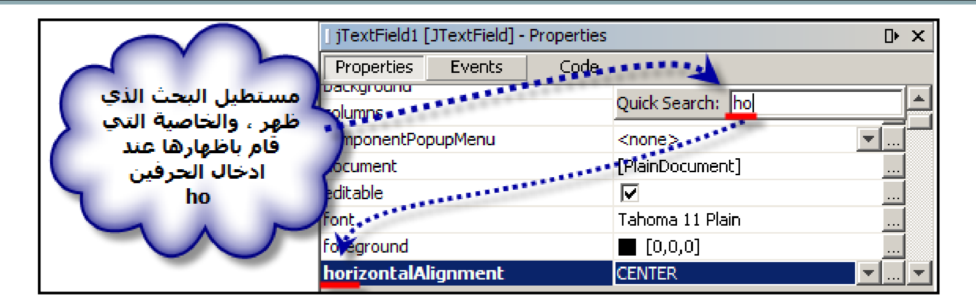


2. ، و عند ادخال القيم و النقر على زر (جمع العددين)، سيظهر بالشكل التالي :



☒ **مهارات :**

◄ للبحث عن خاصيه معينه في نافذة الخصائص ، نقوم بالنقر على شريط الخصائص حتى يتم تفعيل نافذة الخصائص ، ثم نكتب بداية حروف الخاصية التي نريدها عن طريق الكيبورد مباشرة و الذي بدوره سوف يظهر مستطيل بحث مكتوب بداخله الحروف التي تم طباعتها و الذي بدوره سوف يقوم باظهار الخصائص التي تبدا بنفس الحروف المدخله (اي يعمل كعمل الفلتره)، و الشكل التالي يمثل الفكرة :



لتكبير نافذة الخصائص او اي نافذه آخرى في البرنامج لتسهيل التعامل معها نقوم بالنقر المزدوج على شريط العنوان و سوف يتم تكبير اطار النافذه و لتصغيره نقوم بالنقر مرتين مره اخرى .

◄ لغة جافا حساسه لحالة الاحرف (كبيرة او صغيرة) لذلك يمكن تعديل حالة الاحرف و ذلك بتظليل الكود و الضغط على ctrl + space ، بالاضافة انه يقوم باظهار قائمة منسدله تحتوي على دوال نقوم من الاختيار منها و ذلك اما بالنقر على الخيارات التي في القائمه او عن طريق ضغط زر enter .

الفرق بين أنواع البيانات و انواع الكلاسات لاستخدامها عند تحويل البيانات من نصيه الى رقمية :

انواع الكلاسات (Data class)	انواع البيانات (Data type)
Double	**double**
Flout	**flout**
Integer	**int**

◄ انواع المتغيرات :

نوع المتغير	المتغير	الحجم
المتغيرات الصحيحة	byte	8 bit
	short	16 bit
	int	32 bit
	long	64 bit
المتغيرات الكسرية	float	32 bit
	double	64 bit
المتغيرات النصية	char	16 bit
	String	-
المتغير المنطقي	boolean	1 bit

◄ المعاملات الحسابية :

المعامل	الوصف	مثال بلغة جافا
+ - * /	جمع طرح ضرب قسمة	X = A + B X = A - B X = A * B X = A / B
+= -= *= /= %=	جمع ثم إسناد طرح ثم إسناد ضرب ثم إسناد قسمة ثم إسناد باقي القسمة	(A+=B) = (A = A + B) (A-=B) = (A = A - B) (A*=B) = (A = A * B) (A/=B) = (A = A / B) (A%=B) = (A = A % B)
++ -- %	زيادة بمقدار واحد نقصان بمقدار واحد باقي القسمة	(A++) = (A = A + 1) (A--) = (A = A - 1) X = A%B

◄ المعاملات المنطقية :

المعاملات بالشكل الرياضي	شكل العمليات في لغة الجافا	مثال على الشرط	معنى الشرط
=	==	x==y	x تساوي y
≠	!=	x!=y	x لاتساوي y
>	>	x>y	x اكبر من y
<	<	x<y	x اصغر من y
≥	>=	x>=y	x اكبر من او تساوي y
≤	<=	x<=y	x اصغر من او تساوي y
And	& او &&	if (x ==1 & y== 1)	اذا تحقق كلا الشرطين
Or		او ||	if (x = =1 | y== 1)	اذا تحقق احد الشرطين
Xor	^	if (x == 1 ^ y== 1)	-

الجمل الشرطية :

- الجمل الشرطية باستخدام IF :

(1)

```
IF ( الشرط )
{
    ═══ ;  code
}
```

(2)

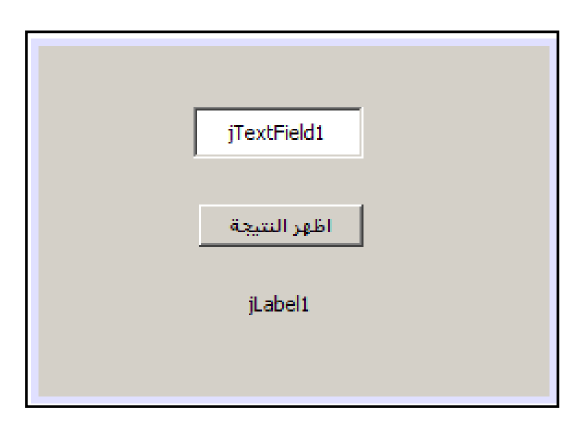
```
IF ( الشرط )
{
    ═══ ;  code
}
Else
{
    ═══ ;  code
}
```

(3)

```
IF ( الشرط )
{
    ═══ ;  code
}
Else IF ( الشرط )
{
    ═══ ;  code
}
Else
{
    ═══ ;  code
}
```

مثال : اكتب برنامج لاظهار نتيجة الطالب بشرط إظهار عبارة ناجح باللون الاخضر و عبارة راسب باللون الاحمر .

قبل التنفيذ :



بعد التنفيذ :



◄ نكتب الكود في الكائن [اظهر النتيجة] :

```java
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
float   m ;
m = Float.parseFloat(jTextField1.getText());
if (m >= 0 & m <50 )
{
jLabel1.setText("راسب");
jLabel1.setForeground(Color.red);
}
else if (m > 50 & m <= 100)
{
jLabel1.setText("ناجح");
jLabel1.setForeground(Color.green);
}
else
{
jLabel1.setText("تاكد من القيم المدخله");
jLabel1.setForeground(Color.black);
}
    }
```

مثال (2) : اكتب برنامج يقوم بتحديد تقدير الطالب باستخدام الـ IF الشرطية .

بنفس الواجهه الموجودة في المثال السابق ولكن كود الكائن [اظهر النتيجة] كالتالي :

```java
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
float   m ;
m = Float.parseFloat(jTextField1.getText());
if (m >= 0 & m <50 )
{
jLabel1.setText("ضعيف");
}
else if (m >= 50 & m < 65)
{
jLabel1.setText("مقبول");
}
else if (m >= 65 & m < 80)
{
jLabel1.setText("جيد");
}
else if (m >= 80 & m < 90)
{
jLabel1.setText("جيد جدأ");
}
else if (m >= 90 & m <= 100)
{
jLabel1.setText("ممتاز");
}
else
{
jLabel1.setText("القيمة المدخله غير صحيحة");
}
    }
```

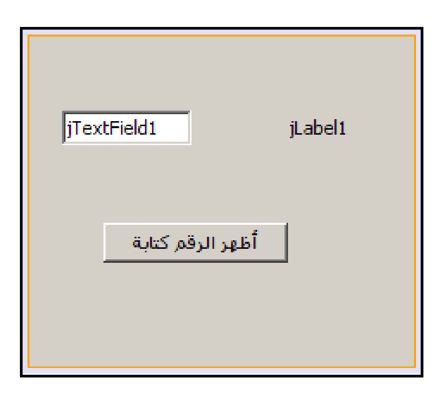
استخدام الدالة Switch :

◄ **الصيغة العامة :**

```
switch ( Test )
{
case value1 :
    ═══ ; code
    break ;

case value2 :
    ═══ ; code
    break ;
    ⋮
default :
    ═══ ; code
    break ;
}
```

◄ **مثال :** أكتب برنامج لإظهار الأرقام بالحروف عند ادخالها بالارقام مثلاً :
(عند ادخل رقم 1 يظهر لنا كلمة one) و هكذا ... .



و لتصغير حجم الكود سيتم ادراج الارقام من الرقم واحد الى الرقم ثلاث و خلاف ذلك سيظهر كتابة تفيد ان الرقم غير مسجل ..

* سيكون كود الزر [أظهر الرقم كتابة] كالتالي :

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
int n = Integer.parseInt(jTextField1.getText());
switch (n)
{
    case 1 :
        jLabel1.setText("one");
        break ;
    case 2 :
        jLabel1.setText("two");
        break ;
    case 3 :
        jLabel1.setText("three");
        break ;
    default :
        jLabel1.setText("Enter the correct number");
        break ;
}

    }
```

2 two
أظهر الرقم كتابة

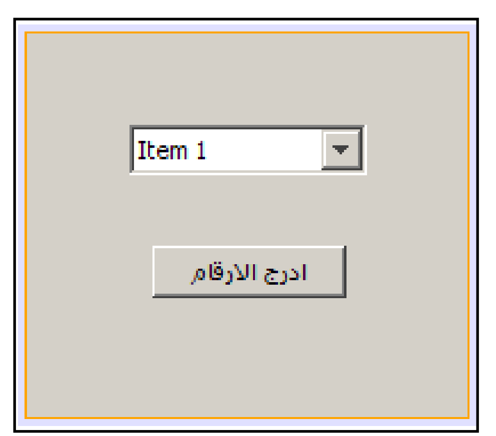
الحلقات (الدوارات) :

1. **الحلقات باستخدام (FOR) :**

◄ الصيغة العامة :

```
for ( نقطة البداية ; الشرط ; مقدار الزيادة )
{
     ; code
}
```

مثال : اكتب برنامج لإضافة الأعداد من واحد الى عشرة الى كائن من نوع JComboBox و سيكون شكل الواجهة بعد ادراج الادوات فيها بالشكل التالي :



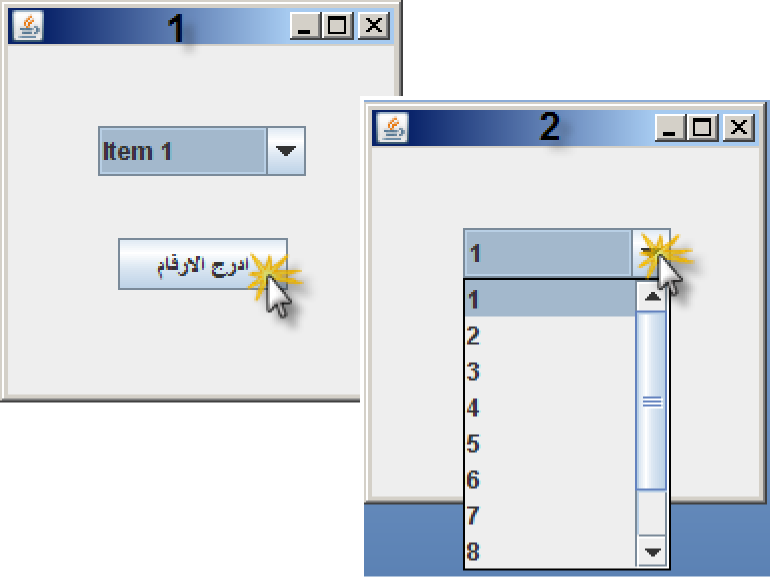
ملاحظة : من أجل إضافة عناصر الى الكائن JComboBox يدوياً ، نتجه الى نافذة الخصائص و نبحث عن الخاصية (model) ، و نقوم بإضافة عناصر بداخل الكائن .

و سيكون كود الزر ادرج الارقام بالشكل التالي :

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
jComboBox1.removeAllItems();
//--> الكود السابق يقوم بتفريغ الكائن من جميع العناصر
for (int i =1 ; i <=10 ; i++)
{
   jComboBox1.addItem(i);
   //--> السطر السابق يقوم باضافة العناصر الى الكائن
}

    }
```

و عند تنفيذ البرنامج يظهر بالشكل التالي :





تدريب على استخدام المهارات :

أكتب برنامج يقوم بايجاد ناتج مضروب عددين بدون استخدام إشارة الضرب .

```
private void jButton1MouseClicked(java.awt.event.MouseEvent
int n1,n2,res = 0 ;
//--> نقوم بتصفير جميع القيم الموجودة في المتغيرات
n1 = Integer.parseInt(jTextField1.getText());
n2 = Integer.parseInt(jTextField2.getText());
for (int i =1 ; i <= n2 ; i++)
{
   res = res + n1 ;
   //--> نقوم باضافة قيمة n1 الثابته الى المتغير res بالاضافة الى قيمته السابقة
}
jLabel1.setText("" + res);
//--> نقوم بطبع النتيجه بشكل نصي لذلك تم دمجها بسلسلة نصية فارغة تمثل علامة """"
    }
```


2. **الحلقات باستخدام (while) :**

◄ الصيغة العامة :

```
while ( الشرط )
{
     ≡;     code
              +
            index
}
```

مثال :

أكتب برنامج لإضافة الأعداد من 20 الى 10 تنازلياً و إضافتها في كائن من نوع **JComboBox**




```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
jComboBox1.removeAllItems();
int x = 20 ;
while (x >= 10 )
{
 jComboBox1.addItem(x) ;
 x--;
}
    }
```

3. الحلقات باستخدام (Do ... while) :

◄ الصيغة العامة :

```
do
{
    ══ ;      code
              +
              index
} while ( الشرط ) ;
```

سؤال : ما الفرق بين الحلقتين الثانيه و الثالثة ؟ في الحلقة الثانية يتم فحص الشرط ثم تنفيذ التعليمات (الكود) ، اما الثالثة سيتم تنفيذ التعليمات ثم فحص الشرط و بالاضافة انه في النوع الثالث سيتم تنفيذ الكود مرة واحدة على الاقل حتى في حالة عدم تحقق الشرط .

مثال : أكتب برنامج لإضافة الأعداد الفردية من 20 الى 1 ، الى كائن من نوع **JComboBox** .

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
jComboBox1.removeAllItems();
int x = 19 ;
do
{
 jComboBox1.addItem(x) ;
 x = x - 2 ;
}while (x >= 1 );
    }
```

Item 1

ادرج الارقام

طريقة إضافة تعليقات الى الكود البرمجي :

- لإضافة تعليقات على الكود البرمجي في سطر واحد يتم كتابة علامة (//) أمام السطر ، كالتالي :

```
jComboBox1.removeAllItems();
 // و نكتب التعليق بهذا الشكل
```

- و لإضافة تعليقات على الكود البرمجي في عدة سطور يتم كتابة علامة (*/) بداية التعليق و يتم كتابة (/*) في نهاية التعليق ، كالتالي :

```
jComboBox1.removeAllItems();
/* write here
 and here */
```

طريقة حفظ المشروع ، بالإضافة الى طريقة فتحة عندنا نريد العمل مرة اخرى على المشروع :

- **طريقة حفظ التعديلات التي تتم على المشروع ، كالتالي :**

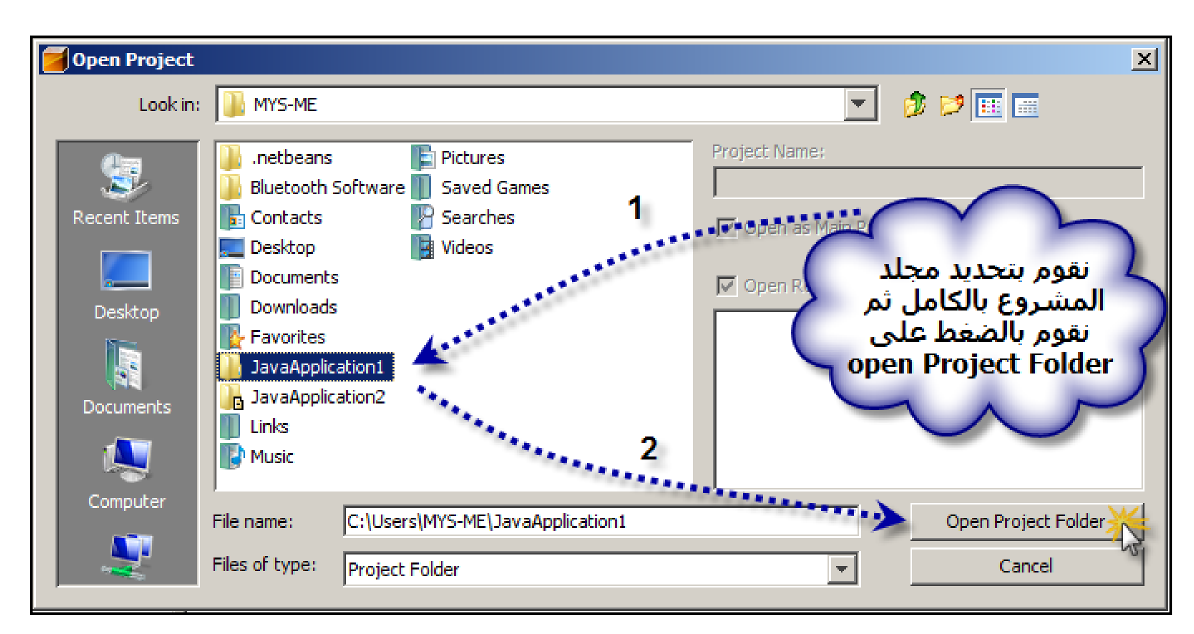
من قائمة File نقوم بالنقر عليها لتظهر لنا القائمة المنسدله و نقوم بالضغط على الخيار save لحفظ الجزء من المشروع الذي يتم عليه العمل او save all لحفظ المشروع بالكامل ، و سوف يتم حفظ التعديلات التي تم إضافتها على المشروع .

ملاحظة : كما يجب التنبيهه انه سوف يتم حفظ التعديلات فقط - التي تمت على المشروع - في المكان الذي تم تحديده مسبقاً ، وذلك عند بداية إنشاء المشروع ، و ليس حفظ ملفات المشروع و موقعها ، و اذا اردنا ان نصل الى مجلد اي مشروع قمنا بتصميمه مسبقاً ندخل المسار الافتراض الذي يقوم البرنامج بحفظ المشروع فيه مباشرةً ، ندخل الى مجلد المستخدم الموجود على سطح المكتب ، و يختلف اسم المجلد من جهاز الى اخر حسب ما تم ادخله اثناء تحميل نظام التشغيل و بجهازي سيكون بهذا المسار : C:\Users\MYS-ME ، و سيكون الاختلاف فقط في MYS-ME .

MYS-ME

- **طريقة فتح المشروع ، كالتالي :**

من قائمة File نقوم بالنقر عليها لتظهر لنا القائمة المنسدله و نقوم بالضغط على الخيار Open Project لفتح المشروع بالكامل ، وسوف تظهر لنا نافذه نقوم من خلالها بعملية تحديد موقع مجلد المشروع بالكامل وليس ملف معين ، ثم نقوم بالضغط على Open project Folder .



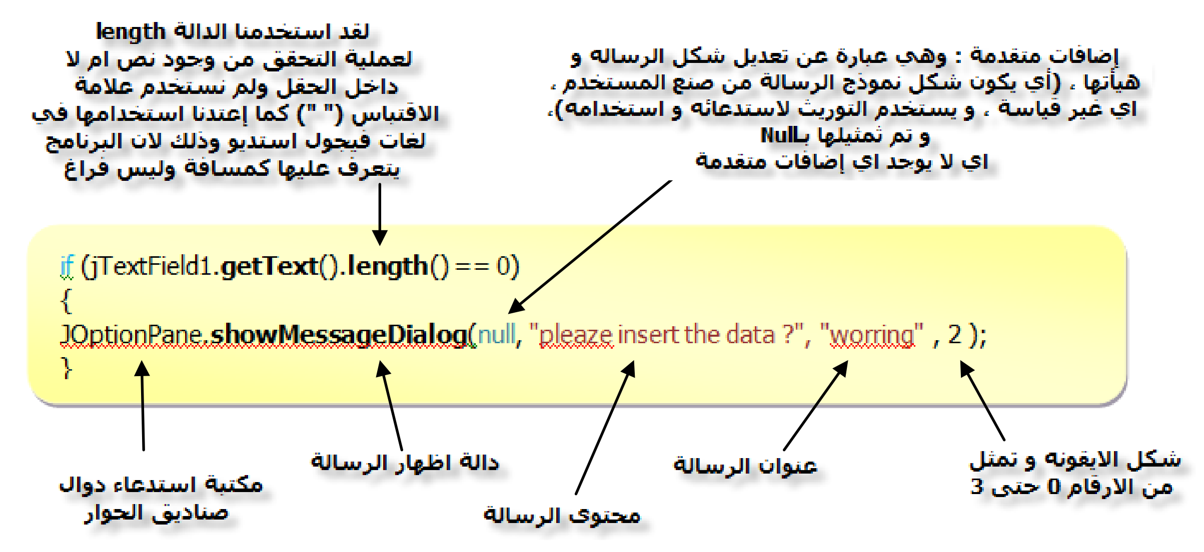
مهارات :
من اجل تنظيم و ترتيب سطور الكود البرمجي في حالة كان هناك سطر متقدم و سطر متاخر ، فاننا نقوم بتظليل السطور المراد تنظيمها ثم نقوم بالضغط على الازرار التاليه في نفس الوقت .
Ctrl + Alt + F

صناديق الحوار و رسائل الإدخال

 **أولاً** : صندوق الحوار Message Dialog :

و هي عبارة عن رسائل تنبيهيه لا أكثر ، بحيث تخبر المستخدم او تقوم بتنبيهه في حالة وجود خطأ في الادخال ، او نسي احد الحقول التي يجب ادخال قيمه فيها ، و كمثال على ذلك :
لو كان لدينا آله حاسبة و قام المستخدم بادخال نصوص في الحقل المراد ادراج قيمة رقميه فيه فقط ، فان البرنامج يرسل للمستخدم رساله يخبره بان هذا الحقل يقبل قيمة رقمية فقط .





 **ثانياً** : صندوق الحوار Confirm Dialog :

و هي عبارة عن رسائل تقوم بتنبيه المستخدم ، ولكن مع وجود تعليمات يتم تنفيذها بناءً على حسب رغبة المستخدم (ازرار تعامل)، و كمثال على ذلك :
لو فرضنا ان هناك بيانات في الحقول ولم يتم حفظها في قاعدة البيانات ، ثم قام المستخدم باغلاق واجهة البرنامج بالخطأ ، فإن الرسالة تظهر امامه و تأكد له عملية الخروج بخيارين ، إما الخروج بدون الحفظ او يقوم بحفظ البيانات ثم الخروج ...الخ .

مثال برمجي : أكتب كود برمجي يقوم بعملية مسح نص مكتوب داخل كائن من نوع jTextField بشرط عن تنفيذ امر المسح تظهر رسالة تأكد من عملية المسح او الغاءه .

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
int sign ;
sign = JOptionPane.showConfirmDialog(null,"Do you want to delete ?","Delete",0,3);
if (sign == 0)
{
jTextField1.setText("");
}
   }
```


القيم الممرة و ماذا ترمز له :

ok و Yes	=	0
No	=	1
cancel	=	2

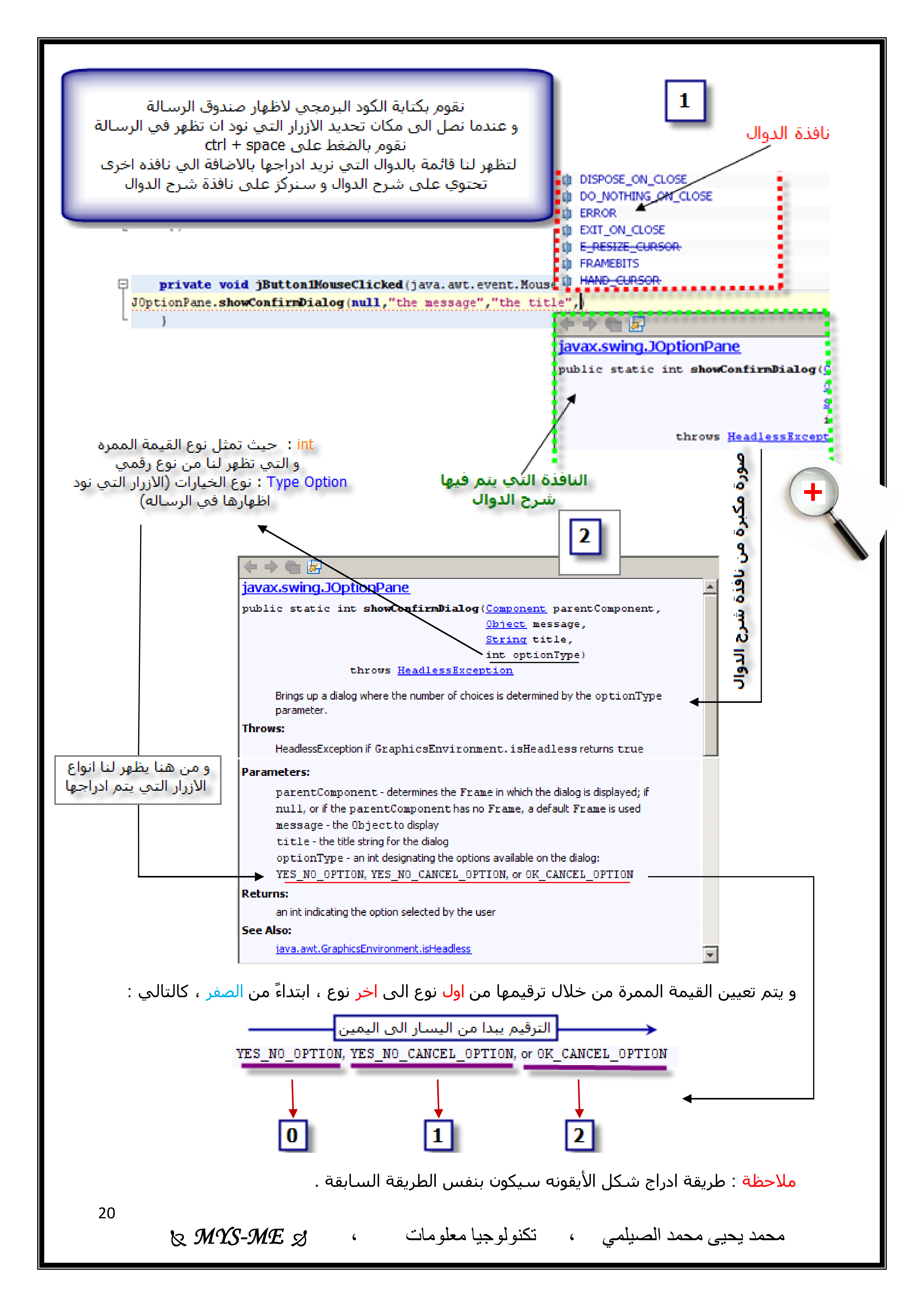
تحديد الازرار التي نريد ظهورها في الرسالة وتمثل قيمها من 0 حتى 2

متغير يقوم باستقبال القيمة الممرة و التي تمثل ما الذي قام المستخدم بضغطه من الازرار اما yes , No , cancel

```
sign = JOptionPane.showConfirmDialog(null,"Do you want to delete ?","Delete", 0 , 3 );
if (sign == 0)
```

عملية المقارنة حيث يقارن القيمة الممرة و القيمة المراد التعامل معها

و يمكن معرفة الازرار التي ستظهر في محتوى الرسالة و الرقم المقابل لها و التي سيتم ادراجه في الكود البرمجي كما ادرجنا الرقم (صفر) و ظهرت لنا الازرار (Yes , No) ، كالتالي :



نقوم بكتابة الكود البرمجي لاظهار صندوق الرسالة
و عندما نصل الى مكان تحديد الازرار التي نود ان تظهر في الرسالة
نقوم بالضغط على ctrl + space
لتظهر لنا قائمة بالدوال التي نريد ادراجها بالاضافة الي نافذة اخرى
تحتوي على شرح الدوال و سنركز على نافذة شرح الدوال

1

نافذة الدوال

DISPOSE_ON_CLOSE
DO_NOTHING_ON_CLOSE
ERROR
EXIT_ON_CLOSE
E_RESIZE_CURSOR
FRAMEBITS
HAND_CURSOR

```
private void jButton1MouseClicked(java.awt.event.Mouse
JOptionPane.showConfirmDialog(null,"the message","the title",)
}
```

javax.swing.JOptionPane
public static int showConfirmDialog(
throws HeadlessExcept

int : حيث تمثل نوع القيمة الممرة
و التي تظهر لنا من نوع رقمي
Type Option : نوع الخيارات (الازرار التي نود
اظهارها في الرسالة)

النافذة التي يتم فيها
شرح الدوال

صورة مكبرة من نافذة شرح الدوال

2

javax.swing.JOptionPane

```
public static int showConfirmDialog(Component parentComponent,
                                    Object message,
                                    String title,
                                    int optionType)
                    throws HeadlessException
```

Brings up a dialog where the number of choices is determined by the optionType parameter.

Throws:

HeadlessException if GraphicsEnvironment.isHeadless returns true

Parameters:

parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is used
message - the Object to display
title - the title string for the dialog
optionType - an int designating the options available on the dialog: YES_NO_OPTION, YES_NO_CANCEL_OPTION, or OK_CANCEL_OPTION

Returns:

an int indicating the option selected by the user

See Also:

java.awt.GraphicsEnvironment.isHeadless

و من هنا يظهر لنا انواع
الأزرار التي يتم ادراجها

و يتم تعيين القيمة الممرة من خلال ترقيمها من اول نوع الى اخر نوع ، ابتداءً من الصفر ، كالتالي :

الترقيم يبدا من اليسار الى اليمين

YES_NO_OPTION, YES_NO_CANCEL_OPTION, or OK_CANCEL_OPTION

0 1 2

ملاحظة : طريقة ادراج شكل الأيقونه سيكون بنفس الطريقة السابقة .

مثال (2) : اكتب كود يقوم باظهار رسالة للمستخدم عند خروجه من البرنامج تفيدة بانه متأكد من الخروج من البرنامج و عند الضغط على yes يتم الخروج و اذا تم ضغط No يعود لنافذة البرنامج .

اولاً : نقوم بتحديد الحدث المناسب لوقت تنفيذ الكود وهو form Window Closing وهو الحدث الذي يتم تنفيذه البرنامج اثناء اغلاق نافذة البرنامج وهو خاص بالكائن Form.

```
private void formWindowClosing(java.awt.event.WindowEvent evt) {
    int r ;
    r = JOptionPane.showConfirmDialog(null,"Do you want to Exit ?","Exit",0,3);
    if (r == 1)
    {
    evt.notify();
    }
}
```

عبارة عن دالة تقوم بايقاف جميع العمليات و الاحداث التي تم تمريرها الى المعالج و الغائها تنفيذها

نقوم بالخروج من البرنامج

Exit

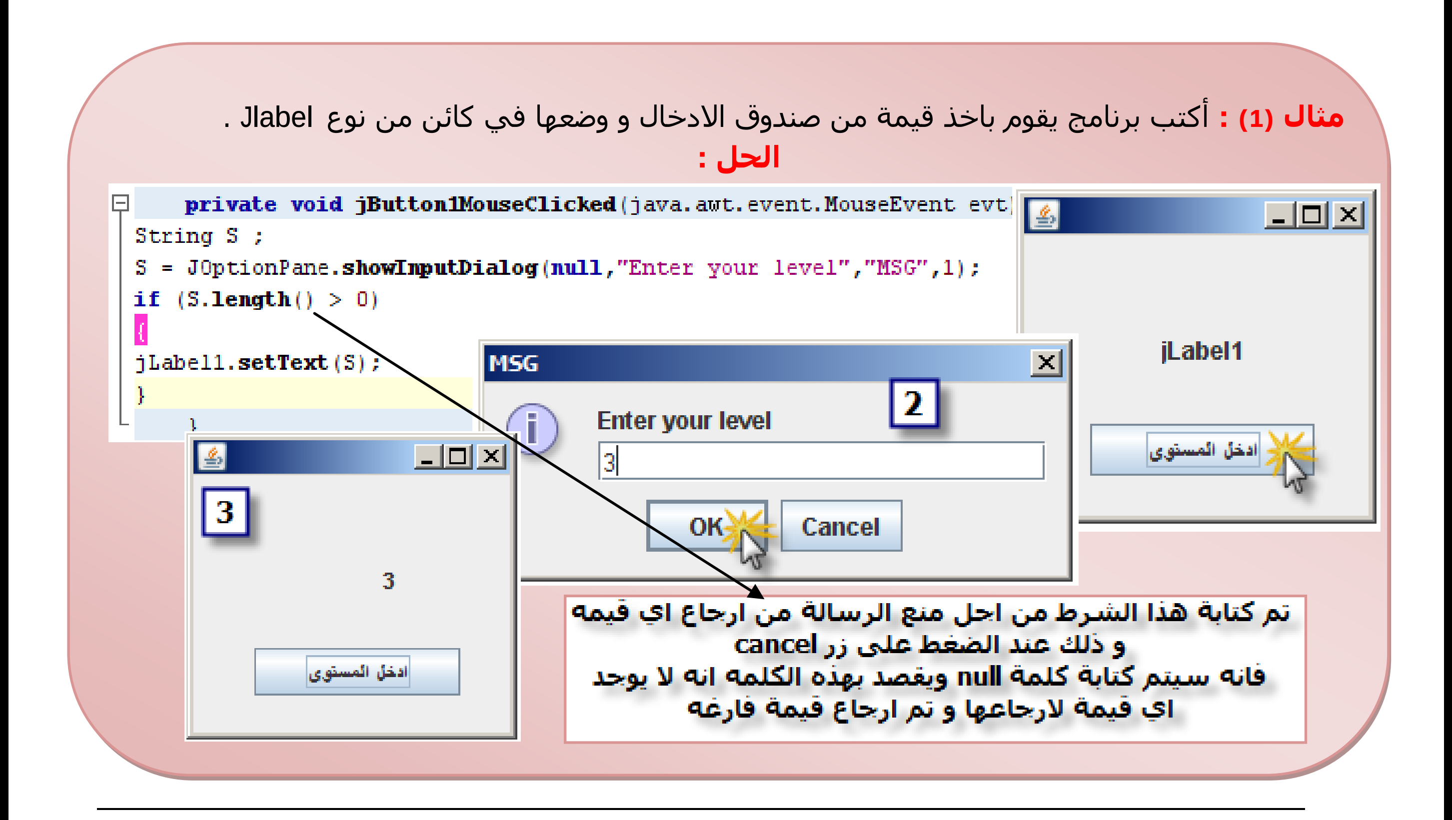
Do you want to Exit ?

Yes No

ثالثاً : **صندوق الإدخال input Dialog :**

الصيغة العامة :

JOptionPane.**showInputDialog() ;**



المصفوفات

الصيغة العامة :

```
Type variable [] = New Type [size];
```

مثال : عرف مصفوفه من نوع عدد صحيح تحتوي على ثلاث خانات .

```
Int x[] = New Int[3] ;
```

كما يجب ان نفرق بين ترتيب العنصر في المصفوفه و عدد عناصر المصفوفه فترتيب عناصر المصفوفه يبدا من الصفر حتى نهاية عناصر المصفوفه وتستخدم في الكود البرمجي للتعامل مع محتوى المصفوفه ، و اما بالنسبه لعدد عناصر المصفوفه تمثل عدد الخانات الموجوده في المصفوفه ، و الشكل التالي يمثل الفرق :



◄ طرق الادخال في المصفوفه -:

1. الطريقة المباشرة :

و تمثل هذه الطريقة ادخال القيم مباشره الى محتوى المصفوفه ، و هي كالتالي :

Type variable[] = **{**… , … , … , … , …**}**;

مثال برمجي : اكتب برنامج يحتوي على مصفوفه و تحتوي المصفوفة على ايام الاسبوع (السبت ، الاحد ، … ، الجمعة) و عند النقر على الزر يقوم باخراج محتوى المصفوفة على كائن من نوع jComboBox .

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
String ary[] = {"saturday","sunday","monday","tuseday","wendseday","thrusday","friday"};
jComboBox1.removeAllItems();
for (int i = 0 ; i < ary.length ; i++)
{
jComboBox1.addItem(ary[i]);
}
    }
```

jButton1

saturday

saturday
sunday
monday
tuseday
wendseday
thrusday
friday

2. طريقة ادراج قيمة من كائن اخر او بطريقة غير مباشرة :

Type variable[] = New **Type** [size];
Variable **[**index**]** **=** value **:**

مثال على طريقة الادخال :

➔ x[0] = 1 ;
➔ x[1] = 5 ;

مثال (1) : اكتب برنامج لترتيب رقمين و إضافة المصفوفه مرتبه في كائن من نوع List .

الحل :

رتب الارقام

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
int arraynumber[] = new int[2];
    int temp ;
    arraynumber[0] = Integer.parseInt( jTextField1.getText() );
    arraynumber[1] = Integer.parseInt( jTextField2.getText() );
    if (  arraynumber[0]  >  arraynumber[1]  )
      {
       temp = arraynumber[0] ;
       arraynumber[0] = arraynumber[1] ;
       arraynumber[1] = temp ;
      }
    for ( int i = 0 ; i <= arraynumber.length ; i++ )
      {
         list1.add( "" + arraynumber[i] );
      }
}
    }
```

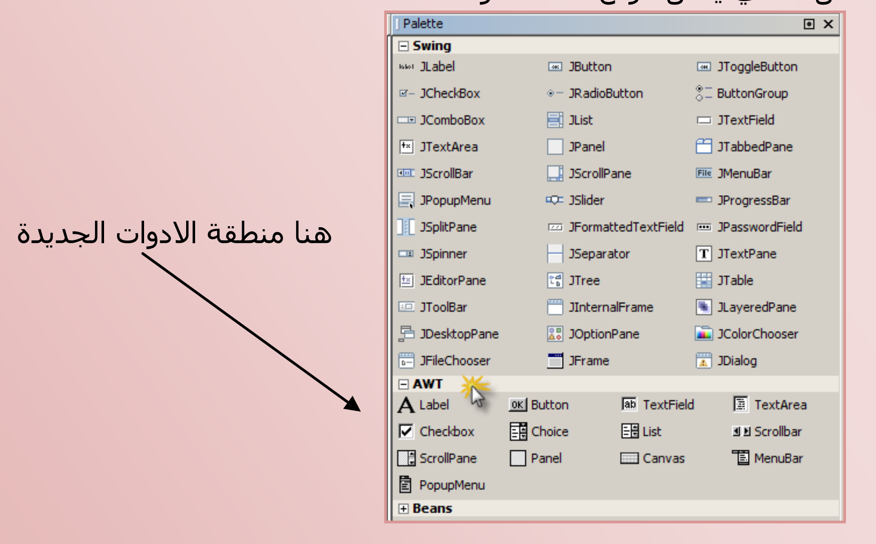
jTextfield1 jTextfield2

رتب الارقام

List1

ملاحظة : هنالك ادوات (كائنات) غير الادوات الظاهره امامنا و من ضمنها list فهي تعتبر من الادوات التابعة لبرامج و تطبيقات الويندوز و يمكن استخدامها في لغة الجافا و يتم الحصول عليها بالطريقة التاليه :

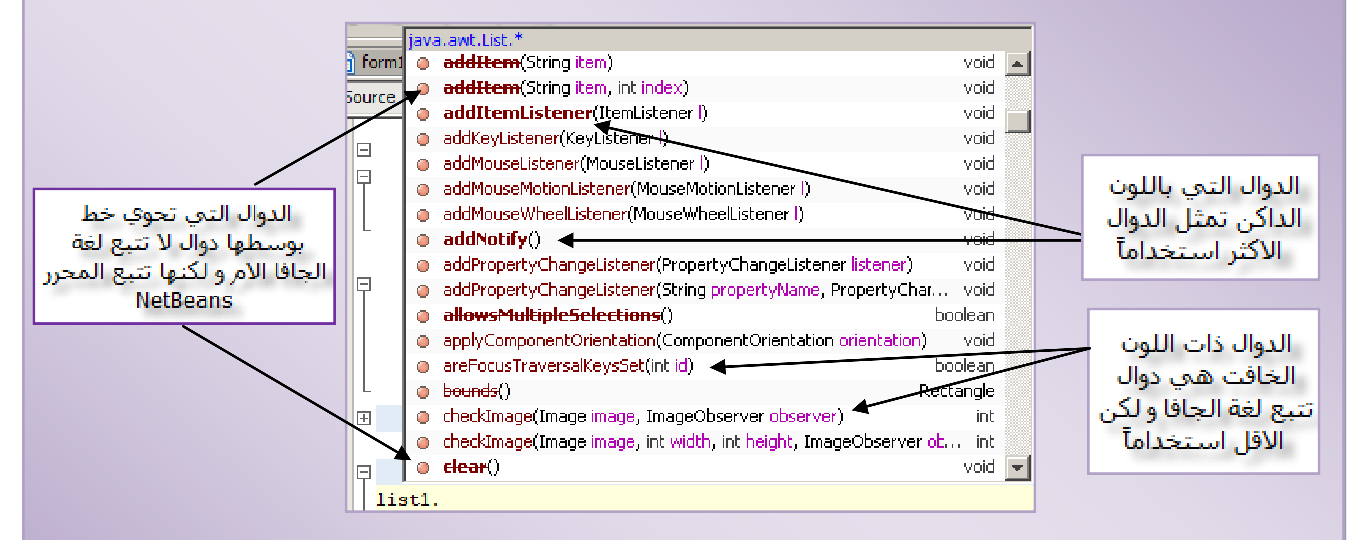
◄ في نفس النافذة الموجود فيها الادوات العادية نقوم بالنزول الي اسفل حتى يظهر لنا شريط عنوان باسم AWT حيث يحتوي على الادوات التابعه لنظام الويندوز و نقوم باظهار محتواه من خلال الضغط على علامة (+) الموجوده بجانبه ، و بعدها يمكن ادراجها على الفورم و استخدامها مثل باقي الادوات ولكن يتم التعامل معها بالدوال الخاصه بها و كما يمكن التعامل معها بالدوال الخاصه بلغة جافا ، و الشكل التالي يمثل موقع هذه الادوات :



ملاحظة : الكائنات الموجودة ضمن AWT تعتبر اخف على النظام ، من الكائنات التي ضمن Swing **ولكن** كائنات AWT تظهر بالشكل الكلاسيكي بالنسبه للكائنات التابعة لـ Swing و التي تظهر بشكل جميل و ثلاثي الابعاد

button1 jButton2

ملاحظه : و من اجل التفريق بين دوال التابعه للغة الجافا و الدوال التابعة للاداة نتبع التالي :
نقوم بادراج كائن list التابع قائمة الادوات الموجوده في AWT و بعد ادراجه في الفورم
نفتح نافذة الاكواد و نقوم بكتابة كود برمجي كالمعتاد و عند الوصول الى اختيار الداله نضغط على Ctrl + Space فتظهر لنا قائمة بحتويات الدوال بهذا الشكل .



و عند اختيار احد الدوال التي تحتوي على خط بوسطها نلاحظ وجود خط اصفر متكسر اسفل السطر البرمجي ، و ذلك يعني ان الدوال المستخدمه دوال خاصه بالمحرر NetBeans و ليست خاصة بلغة جافا ، و الصورة التاليه توضح الخط الاصفر :

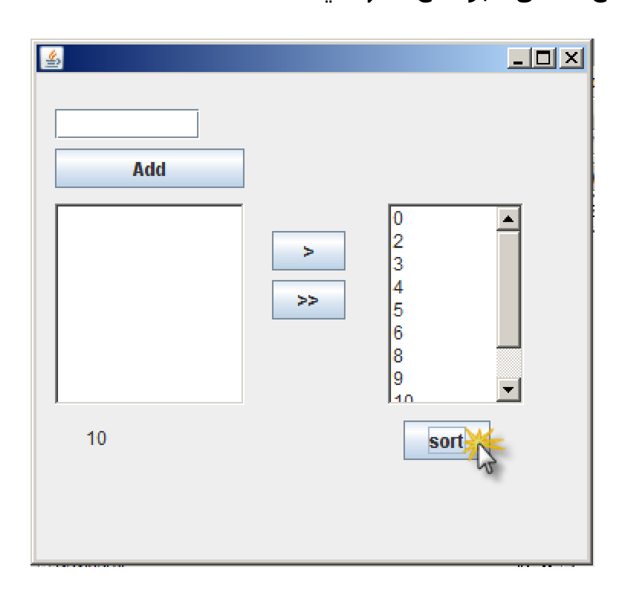
```
list1.addItem("jbj");
```

```
list1.clear();
```

و لكن عند اختيار احد الدوال التابعه للغة الجافا فان الكود البرمجي يظهر بالشكل العادي :

```
list1.add("jbj");
```

مثال (2) : اكتب برنامج يقوم باضافة عشرة عناصر (ارقام) الى كائن من نوع list ثم يقوم المستخدم بارسال المدخلات الى قائمه اخرى (list) و يقوم بالضغط على زر ترتيب ثم يقوم بترتيب محتوى القائمة ، و الشكل التالي يوضح شكل البرنامج النهائي :



الحل :

sort

```java
private void jButton4MouseClicked(java.awt.event.MouseEvent evt) {
 int x[] = new int[10];
 for (int y = 0 ; y <= list2.getItemCount() -1; y++)
{
 x[y] = Integer.parseInt(list2.getItem(y)) ;
}
 list2.removeAll();
// int x[] = {5,6,9,4,3,2,8,7,5,1};
int temp ;
for (int i = 0 ; i <= x.length -2; i++)
{
   for (int j = i + 1 ; j <= x.length - 1; j++)
   {
       if ( x[i] > x[j])
       {
       temp = x[i];
       x[i] = x[j];
       x[j] = temp ;
       }
   }
}
 for (int m = 0 ; m <= x.length -1; m++)
{
 list2.add( "" + x[m]);
}
    }
```

>

```java
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
     list2.addItem(list1.getItem(list1.getSelectedIndex()));
     list1.remove(list1.getSelectedIndex());
 }
```

Add

```java
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
  if (jTextField1.getText().length()==0)
    {
      JOptionPane.showMessageDialog(null,"Enter number","Error",2);
    }
    else
    {
      list1.addItem(jTextField1.getText());
      jTextField1.setText("");
      Lbl.setText("" +list1.getItemCount());
    }
 }
```

>>

```java
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {

     for (int i=0 ;i <list1.getItemCount() ; i++)
     {
         list2.addItem(list1.getItem(i));
     }
     list1.removeAll();
 }
```

الواجهات و الاحداث

التوريث :
و هي عبارة عن استدعاء كائن جديد بنفس صفات الكائن الذي تم توريث الصفات منه .

← طريقة (عملية) توريث و استدعاء فورم جديد له نفس خصائص الفورم المستدعى منه :-
1. نفتح مشروع جديد و نقوم بادراج زر في الفورم و كتابة الكود البرمجي التالي :

حيث تمثل اسم الفورم المراد توريث صفاته للفورم الجديد

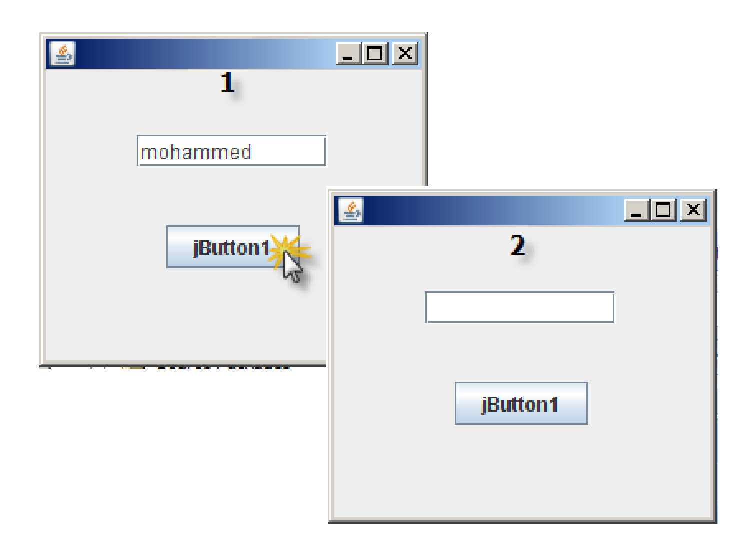
```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
Form1 f2 = new Form1();
f2.show();
this.hide();
    }
```

اسم الفورم الجديد

السطر الثاني :
يمثل عملية اظهار الفورم الجديد و كما نلاحظ اسفل السطر خط اصفر متكسر و ذلك يدل على ان hide دالة تابعة لنظام الويندوز .

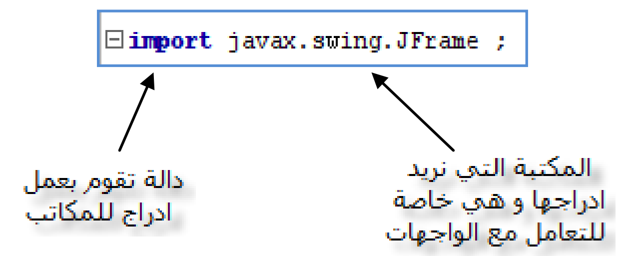
السطر الثالث :
يمثل عملية اخفاء للفورم السابق و كلمة this دالة تعود على الفورم الذي يتم العمل عليه و هو في هذه الحالة يمثل Form1.

◄ و من اجل ملاحظة الفرق بين الفورم الذي سيتم ظهوره و الفور القديم ، يتم ادراج كائن من نوع jTextField و عند تنفيذ الكود البرمجي يتم كتابة اي نص في الفورم القديم ثم الضغط على الزر فنلاحظ اختفاء النص الذي تم كتابته و ذلك يدل على ظهور فورم جديد غير السابق و لكن بنفس الشكل ، و الصورة التالية توضح الفرق :



◄ طريقة آخرى ولكن عن طريق ادراج مكتبة الفورمات بالاضافة لكتابة الكود داخل المشيد :

1. نقوم بادراج مكتبة التعامل مع الواجهات ، حيث يتم كتابة الكود بين package و قبل public class و كود ادراج المكتبة كالتالي :



2. نذهب الى الاجراء التالي و نمسح الكود من داخله :

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Form1().setVisible(true);
        }
    });
}
```

و نقوم بكتابة الكود التالي :

```
public static void main(String args[]) {
    Form1 x = new Form1();
    x.show() ;
}
```

← و سيكون الشكل العام لنافذة الاكواد بالشكل التالي :

```
/**
 *
 * @author  MYS-ME
 */
import javax.swing.JFrame ;
public class Form1 extends javax.swing.JFrame {

    /** Creates new form Form1 */
    public Form1() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    Generated Code

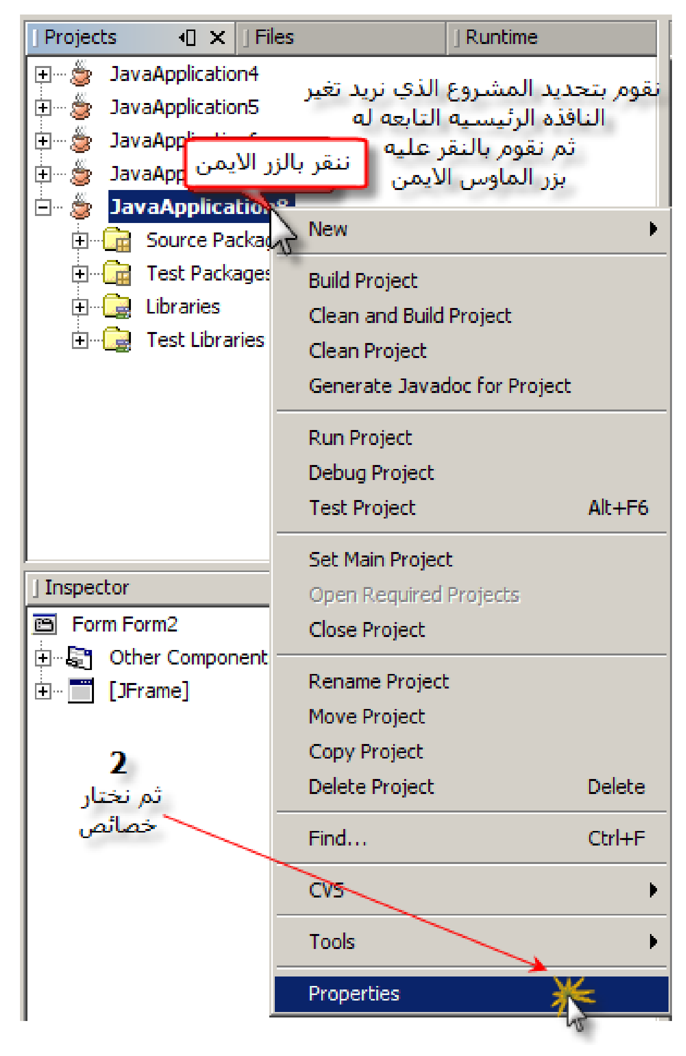
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        Form1 x = new Form1();
        x.show() ;
    }
```

◄ الكود البرمجي لزر الخروج (اغلاق البرنامج) End ، كالتالي : `System.exit(0);`

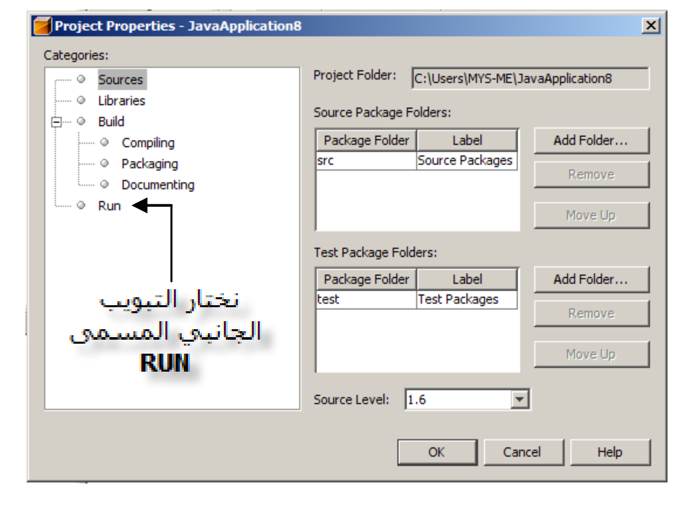
◄ طريقة تحديد الواجهة الاولى التي ستظهر للمستخدم عند بداية تشغيل البرنامج :

1. في البداية نتبع الاتي :

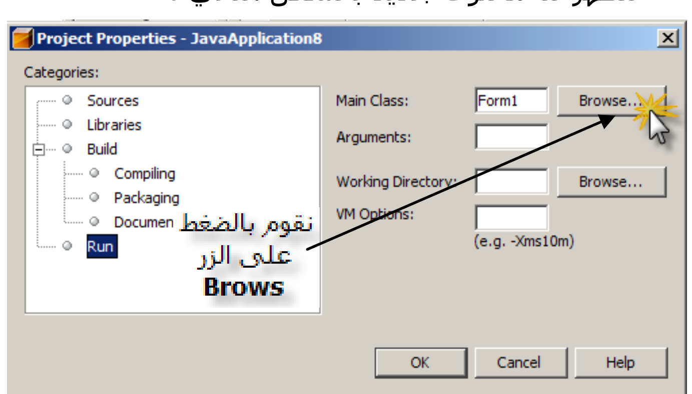




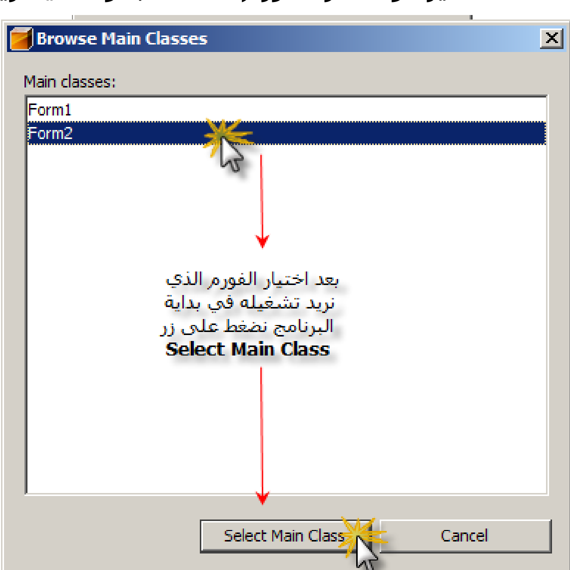
2. و سوف تظهر نافذة جديدة بالشكل التالي :



لتظهر لنا محتوى جديد بالشكل التالي :



3. و بعد ذلك تظهر لنا النافذة الاخيرة و نختار الفورم المناسب و الذي نريد تشغيله في بداية البرنامج :



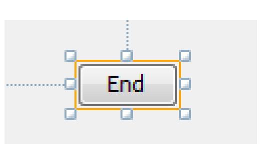
تنبيه : من اجل التاكد من عملية الاختيار نقوم بتفيذ البرنامج و نلاحظ ظهور الفورم الثاني

← الأحداث : -

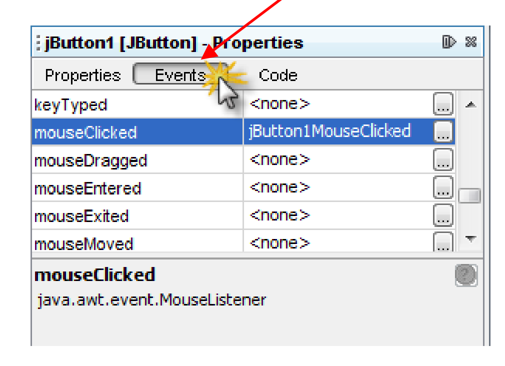
1. إضافة حدث :

يمكن اضافة عدة أحداث تحت نفس الحدث الرئيسي و سوف يقوم بتنفيذها بالتتابع .
و الطريقة لفعل ذلك كالتالي :

1. نحدد الكائن المراد اضافة حدث اليه (عن طريق النقر عليه) .

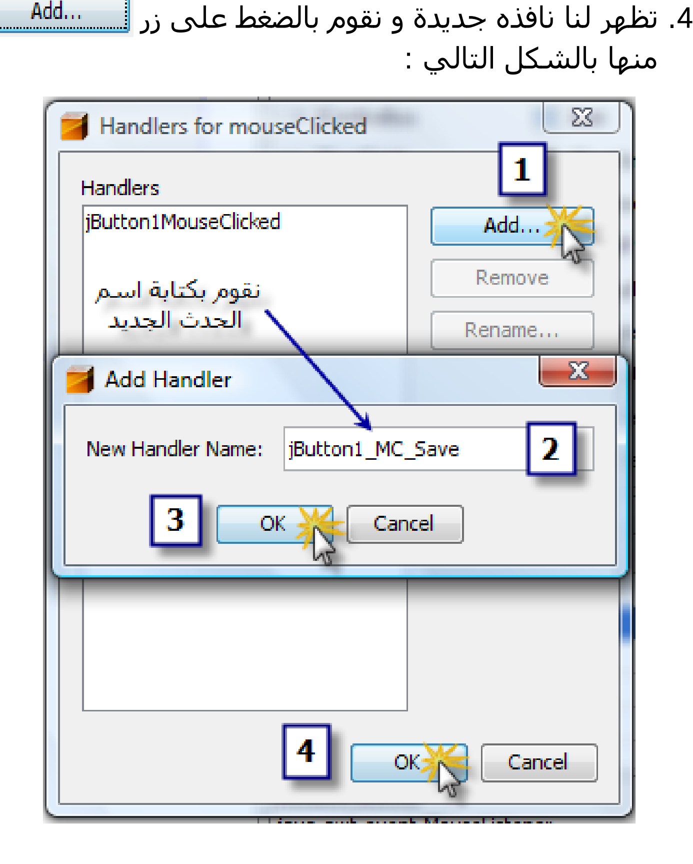


2. نذهب الى نافذة الخصائص و نختار التبويب Event .



3. نحدد الحدث الذي نريد اضافة حدث اخر بداخله و نقوم بالضغط على الزر ... الذي امام الحدث .

4. تظهر لنا نافذة جديدة و نقوم بالضغط على زر Add... منها بالشكل التالي :



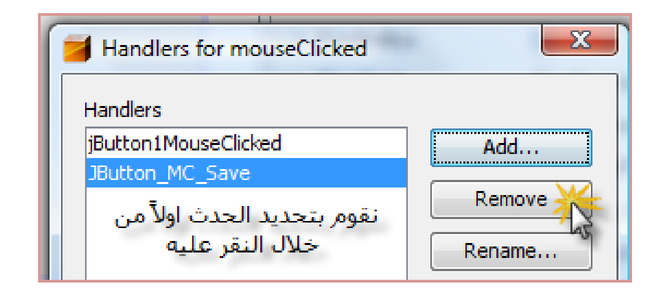
◄ و بعد الانتهاء من اضافة الحدث سيظهر لنا الحدث في نافذة الاكواد لنكتب الشفرة التي نريد تنفذها ، و الصورة التالية توضح لنا ظهور الحدث الجديد الذي قمنا باضافته :

```
private void jbutton_mc_save(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
    }

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
System.exit(0);
    }
```

2. حذف حدث :

من اجل حذف اي حدث يتم اتباع الخطوات السابقه في اضافة حدث ولكن عند الوصول النقطه الرابعه يتم تحديد الحدث الذي نريد حذفه من القائمه ، و نضغط على زر Remove :



ملاحظة :
و من أجل تغيير اسم الحدث نتبع نفس الطريقة و لكن نضغط على الزر Rename بدلاً من الزر Remove

3. إستدعاء حدث من حدث اخر :

و ذلك من خلال وضع اسم الحدث الذي نريد تطبيقه ضمن كود الحدث الآخر .

◄ مثال: يوجد لدينا كود معين و بمجرد الضغط على مفتاح Enter من لوحة المفاتيح يتم استدعاء حدث اخر .

الحل :

اولاً نختار الحدث المناسب وهو يمثل في هذه الحاله الحدث (Key pressed) :

ثم نقوم باضافة الكود الذي نريد تنفيذه ، كالتالي :

ارقام اسكي و رقم عشرة يمثل الزر Enter من لوحة المفاتيح

```
private void jButton1KeyPressed(java.awt.event.KeyEvent evt) {
if (evt.getKeyCode() == 10 )
{
jButton1MouseClicked(null);
}
    }
```

متغير يمثل لنا القيمة الممرة الناتجة من ضغط لوحة المفاتيح

الحدث الذي نريد استدعائه من هذا الحدث

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
System.exit(0);
    }
```

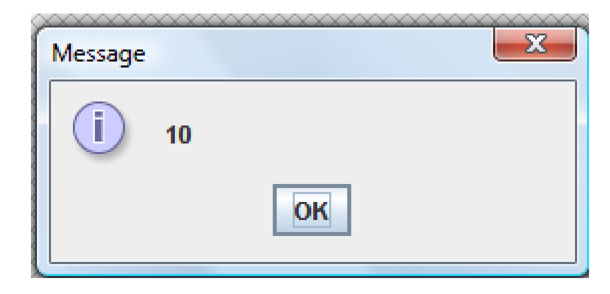
الحدث الذي تم استدعائه و الذي بدورة سيتم تنفيذ الكود الخاص به و الذي سيقفل البرنامج

ملاحظه : في حالة عدم كتابة الشرط if في الكود السابق فان الحدث سوف يتم تنفيذه عند الضغط على اي زر من لوحة المفاتيح .

مهارات : من اجل معرفة رقم الكي اسكي الذي يقابل (يمثل) اي زر ، نكتب الكود التالي :

```
private void jButton1KeyPressed(java.awt.event.KeyEvent evt) {
JOptionPane.showMessageDialog(null , evt.getKeyCode());
    }
```

و عند تنفيذ البرنامج و الضغط على زر Enter يظهر لنا الرقم الكي اسكي الذي يمثل الزر ، بالشكل الاتي :



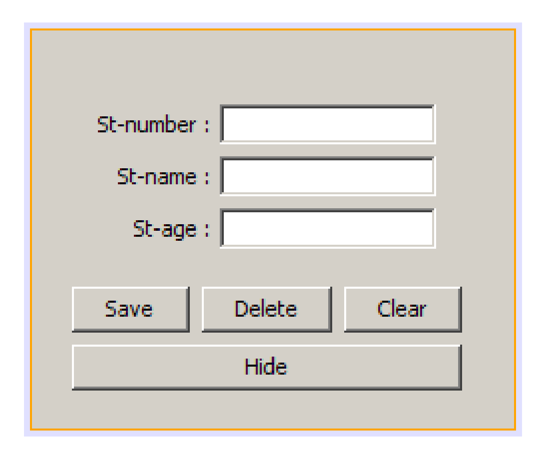
الدوال و الإجراءات

فوائد الدوال و الاجراءات :

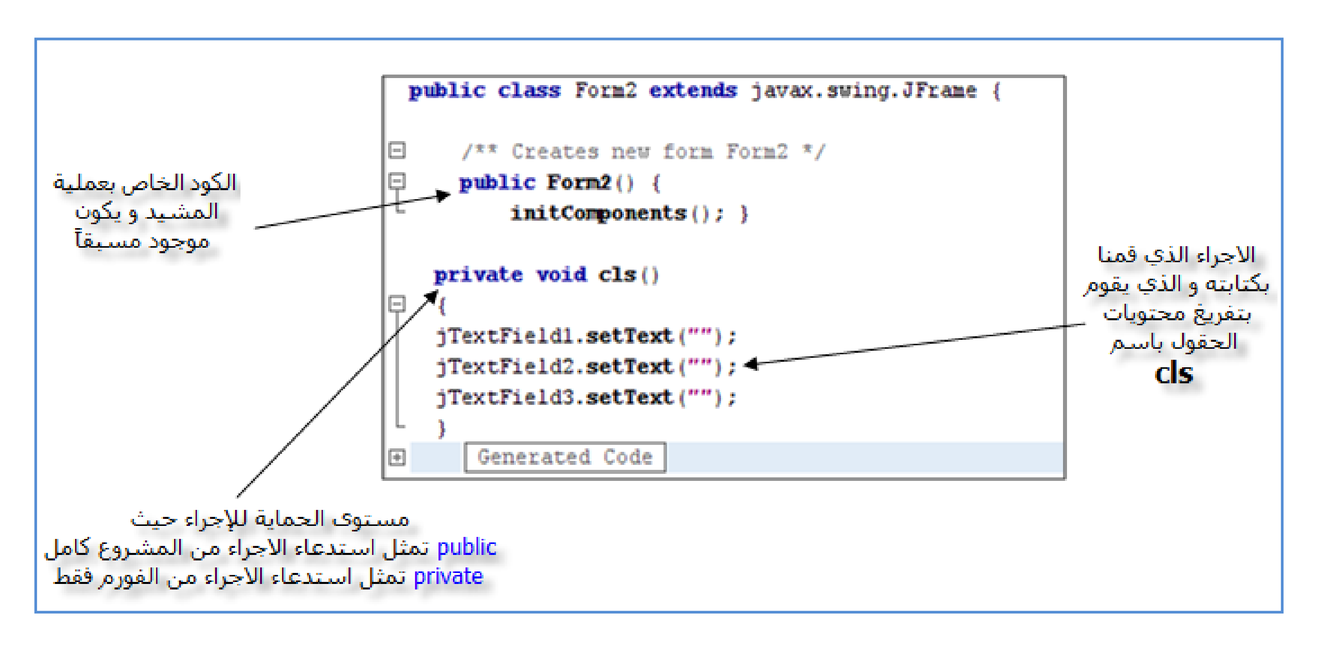
1. إختصار الشفرة البرمجية (بحيث لا تتكرر في اكثر من زر) .
2. ترتيب الشفرة البرمجية .
3. سهولة معالجة و اكتشاف الاخطاء .
4. كتابة وظائف جديدة .

اولاً : الإجراءات (Procedure) : و هي لا ترجع قيمة .

مثال(1) : على سبيل المثال يوجد لدينا برنامج مرتبط بقواعد البيانات بحيث هذا البرنامج يحتوي على ازرار (الحذف و الحفظ و تفريغ) بحيث عند الضغط على تفريغ يتم مسح جميع الحقول و عند الضغط على حفظ فانه يقوم بالحفظ في قاعدة البيانات و لكن يقوم ايضا بعملية تفريغ للحقول بعد حفظها ، و شكل البرنامج كالتالي :



نقوم اولاً بكتابة الإجراء الذي نريد تنفيذه ، و الذي سيكون موقع تعريفه بعد المشيد بالطريقة التالية :



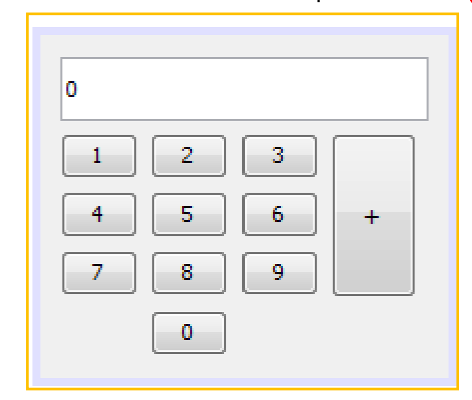
و من اجل استدعاء هذا الاجراء من زر معين نكتب الكود البرمجي التالي ضمن كود الزر :

```
cls();
```

و نقوم بكتابة الكود السابق ضمن كود زر الحفظ بالاضافة الى زر تفريغ الحقول ، بالشكل التالي :

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
cls();
    }
```

مثال (2) : **آله حاسبة -:**



مهارة : من اجل اضافة عدة كائنات من نفس النوع الى الواجهه بدون الجوء في كل مره الى نافذة الادوات ، حيث يتم اختيار الكائن من نافذة الادوات و قبل ادراجه في الواجهه يتم الضغط مع الاستمرار على الضغط على زر shift ثم يتم ادراج الكائن على الواجهه من خلال النقر على الواجهه حسب الرغبه من عدد الكائنات التي نريد ادراجها ، و عند الانتهاء يتم فك الضغط من على الزر Shift .

ملاحظة : يجب ادراج قيمة الصفر في الحقل jTextField كقيمة افتراضه من خلال الخاصية Text ، كما هو وجود في الصورة السابقه .

◄ من اجل منع الكتابة على الكائن jTextField نقوم بالغاء الصح من مربع الخيار الموجود امام الخاصية Focusable كما في الصورة التالية :



focusTraversalPolicyProvider	☐
focusable	☐
highlighter	[BasicHighlighter]

1. نقوم في البداية بكتابة الإجراء الخاص بالآله الحاسبه من اجل استدعائه في باقي الازرار : وسيكون الكود او الاجراء الذي نريد تعريفه بعد كود المشيد كما ذكرنا سابقاً :

عملية دمج الناتج مع نص لوضعه في المتغير الجديد الذي من نوع نصي

داله رياضية تقوم بتقريب الرقم الكسري الى اقرب رقم صحيح من اجل التخلص من الفاصله الموجوده في الرقم

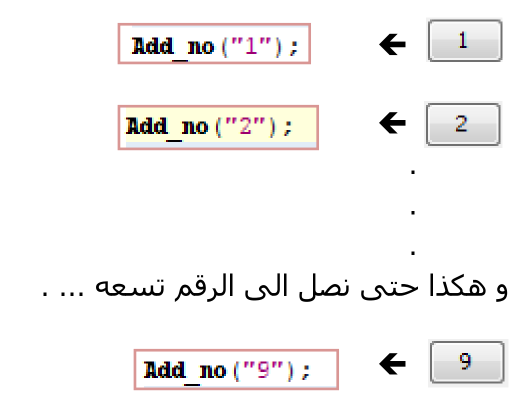
من اجل تحويل القيمة النصيه الى قيمة رقمية من نوع عدد كسري

```
public void Add_no(char N)
{
String old_no = "" + Math.round(Double.parseDouble(jTextField1.getText()));
String New_no = old_no + N ;
Double d = Double.parseDouble(New_no);
jTextField1.setText("" + d );
}
```

عملية دمج الرقم الممرر مع الرقم القديم الموجود في jtextfield

هذا السطر من اجل التخلص من الصفر الذي ياتي في بداية الرقم حتى اذا كان الرقم بهذا الشكل (01) يعدل الرقم و يصبح (1) فقط بدون الصفر الذي في البداية

و اجل تمرير قيم الى الاجراء السابق ، نكتب الاكواد التاليه في كل زر بالشكل التالي :

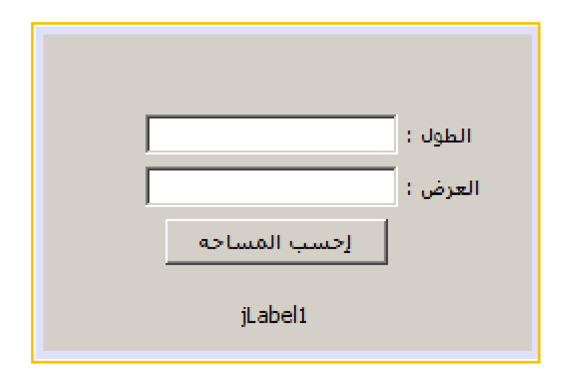


و هكذا حتى نصل الى الرقم تسعه ... .

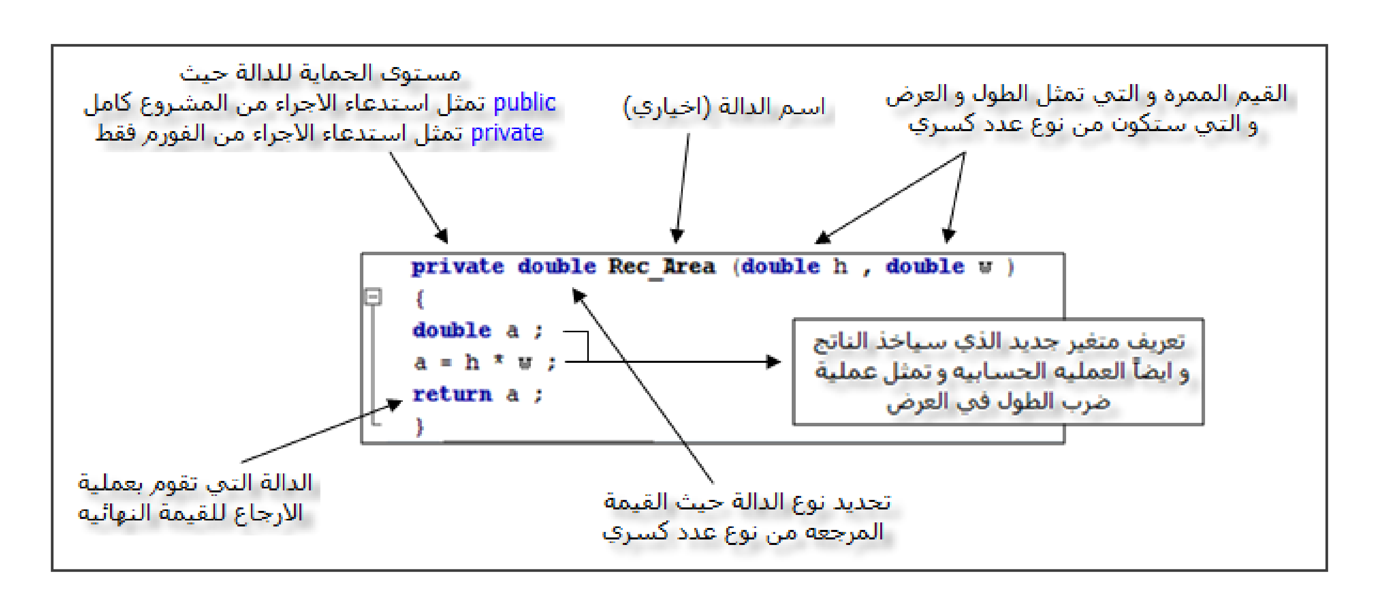
9 ← Add_no("9");

ثانياً : الدوال (Function) : حيث تقوم بارجاع قيمة بعكس الاجراء .

مثال (1) : اكتب برنامج يحتوي على **دالة** تمرر اليه قيمة الطول و العرض و يعيد مساحة المستطيل :



نقوم بكتابة الدالة بعد المشيد مباشرةً كما كنا نفعل في الإجراء :



و في الزر إحسب المساحه نقوم بكتابة الكود التالي :

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
double h1 , w1 ;
h1 = Double.parseDouble(jTextField1.getText());
w1 = Double.parseDouble(jTextField2.getText());
jLabel1.setText("" + Rec_Area(h1 , w1 ));
    }
```

الطول : 5
العرض : 6
إحسب المساحة
30.0

طريقة تمرير القيم الى الداله بالشكل المقابل

مثال (2) : اكتب برنامج يحتوي على مصفوفه تحتوي على عشرة اسماء ، و **دالة** تستخدم للبحث عن هذا الاسم الممر اليها، و يعيد (Name found) في حالة الاسم موجود ، او (Name not found) في حالة اذا لم يكن موجود .

1. نقوم بتعريف مصفوفه عامه ، حيث يتم تعريفها خارج المشيد ، و خارج اي اجراء او داله :

```
String ary[] = {"ali","mohammed","ahmed","hussen","khaled","omer","adel","nasser","tareq","bader"};

private boolean F_name(String N)
{
    for (int i=0 ; i<ary.length;i++)
    {
        if (N.equals(ary[i]))
        {
            return true;
        }
    }
    return false;
}
```

2. نقوم بانشاء داله من نوع منطقي بحيث تعود لنا إما بقيمة True او False و الشكل المقابل يمثل انشاء الدالة .

Equals : هي عبارة عن دالة تستخدم للمقارنه بين اصناف مختلفه من البيانات مثل المقارنه بين مرصوفه و محتوى متغير او بين حقل Text و متغير مثل عملية ادخال الرقم السري و اسم المستخدم في حقل text و مقارنته هل هو صحيح ام لا .

... الدالة بالشكل التالي:

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
        if (F_name(jTextField1.getText())== true)
          JOptionPane.showMessageDialog(null,"Name found");
      else
          JOptionPane.showMessageDialog(null,"Name not found");
    }
```

عندما يجد الاسم ستظهر هذه الرساله

ali
إبحث
Message
Name found
OK

و عندما لا يجد الاسم يظهر هذه الرساله

yasser
إبحث
Message
Name not found
OK

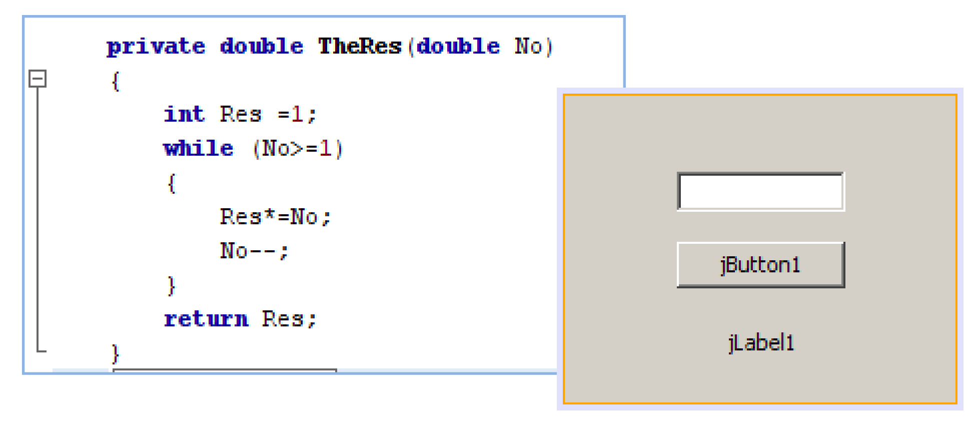
ملاحظة : كما لاحظنا من كتابة الدالة في المثال السابق و الذي يمثل الكود الذي بالمقابل ← انه في اي داله تقوم بكتابتها يجب ان تذكر الكلمة المفتاحيه (**return**) - والتي تعيد القيم- ، مره واحده على الاقل و تكون خارج الجمل الشرطيه و جمل التكرار .

```
private boolean F_name(String N)
{
    for (int i=0 ; i<ary.length;i++)
    {
        if (N.equals(ary[i]))
        {
            return true;
        }
    }
    return false;
}
```

مثال (3) : اكتب برنامج يحتوي على دالة (**Prosegur**) يمرر اليه الرقم و يعود بضروبه .

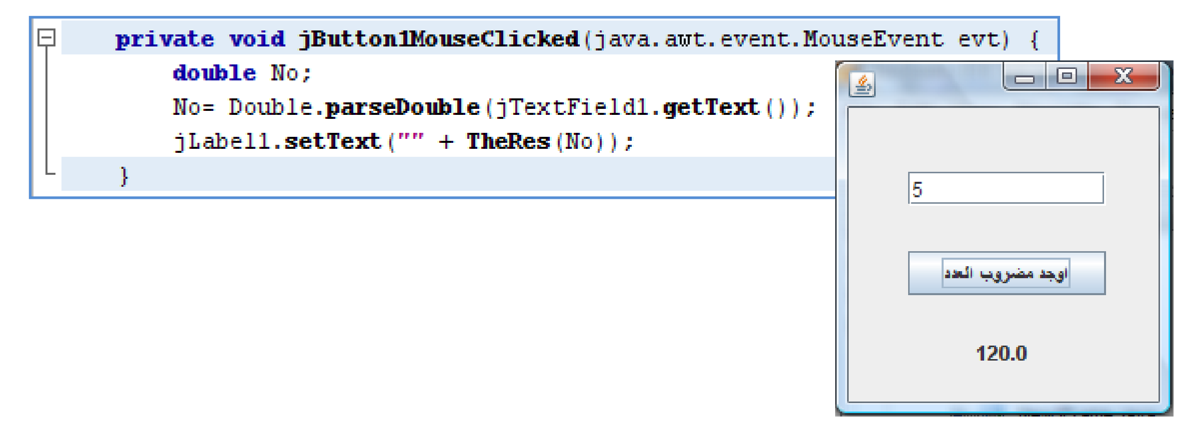
الحل :

نقوم بتعريف دالة من نوع عدد كسري بالشكل التالي :



```
private double TheRes(double No)
{
    int Res =1;
    while (No>=1)
    {
        Res*=No;
        No--;
    }
    return Res;
}
```


و كود زر التنفيذ كالتالي :



```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    double No;
    No= Double.parseDouble(jTextField1.getText());
    jLabel1.setText("" + TheRes(No));
}
```


مهارة : طريقة تغيير الاسم البرمجي للكائن .

و ذلك من الخلال النقر بزر الماوس الايمن على الكائن المراد تغيير اسمه البرمجي و تظهر لنا قائمة منسدله لنختار منها Change Variable Name ... لتظهر لنا نافذة جديدة نقوم من خلالها بكتابة الاسم البرمجي الجديد للكائن ثم نضغط على الزر OK .

التعامل مع قواعد البيانات

مفاهيم اساسية :

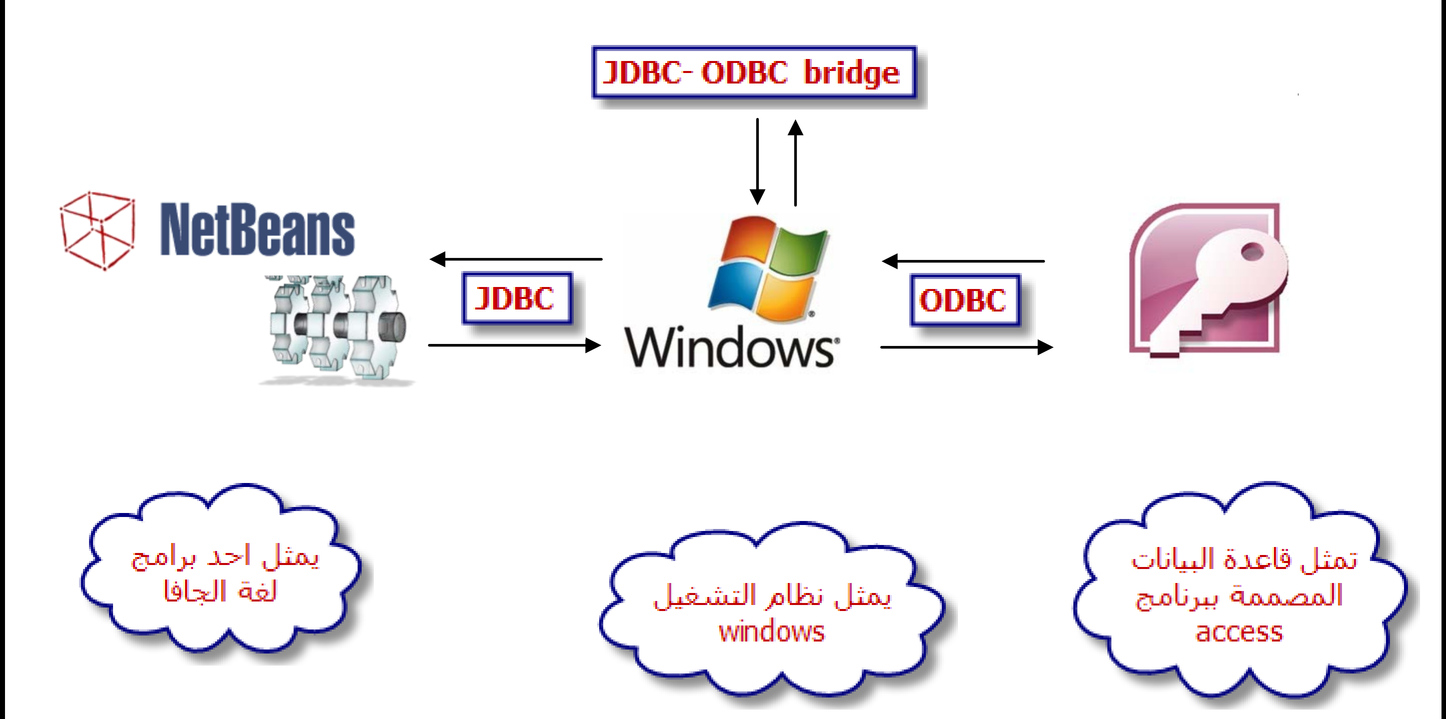
◄ في لغة جافا لا يمكن ربط قاعدة بيانات مباشرةً مع اي تطبيق مكتوب بلغة الجافا، ولذلك اذا اردنا ان نربط ، فإننا نحتاج الى السرواقات (Driver) الموجوده داخل نظام التشغيل (ODBC) ولذلك هناك بعض المصطلحات التي سنستخدمها في عملية الربط ، من ضمنها :

← JDBC وهو اختصار لـ (Java Database Connectivity) :
وهو يأتي مع الـ JDK ، فائدته أنه يسمح لك أن تستخدم لغة (SQL) وهي اللغة المستخدمة للاتصال بأي قاعدة بيانات سواء كانت MySQL او Access او Oracle ، بحيث يقوم بربط بين التطبيقات المصممة بلغة جافا مع نظام التشغيل .

← ODBC وهو اختصار لـ (Open Database Connectivity) :
يقوم نظام التشغيل بفتح اتصال مع قاعدة البيانات التي سنقوم بتحديدها ، بشكل مبسط عملية ربط قاعدة البيانات مع نظام windows .

← (JDBC- ODBC bridge) :
هنا تأتي فائدة الـ (JDBC-ODBC bridge) حيث أنه يقوم بترجمة اوامر الـ JDBC الى اوامر ODBC ، (اي عامل ربط بين الاثنين السابقين) .

◄ **و الشكل التالي يمثل شكل عملية الربط :**

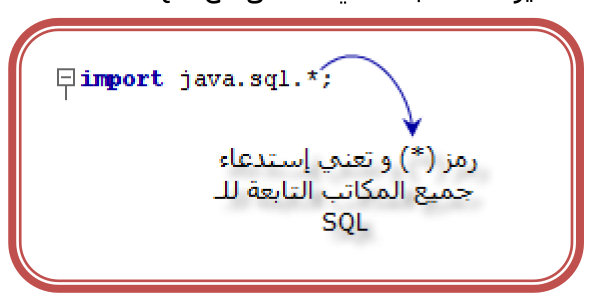


◄ لربط برنامج مصمم بلغة جافا مع سواقات Driver الموجودة داخل نظام التشغيل (PDBC) تتبع الخطوات التالية :

1. استدعاء مكتبة الربط و المشغلات :

أ. إستدعاء مكتبة الربط الخاصة بالتعامل مع جمل SQL .

➔ طريقة و كود استيراد المكتبة التي تتعامل مع SQL :



ملاحظه : يتم كتابة اي استدعاء او استيراد للمكاتب خارج المشيد باعلى **public class** .

ب. إستدعاء المشغل الخاص بالتعامل مع قواعد البيانات من الجافا .

س

2. ربط قاعدة البيانات مع البرنامج باستخدام الكائنات :

أ. إنشاء رابط (**Connection**) .

ب. إنشاء جمل التحكم (**Statement**) .

ت. إنشاء مخزن للبيانات (**Result Set**) .

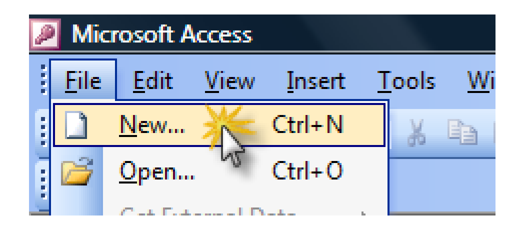
ملاحظة : جميع اوامر قواعد البيانات يجب ان تكتب داخل جملة تصيد الاخطاء حتى و ان كانت الاوامر صحيحه ، و صيغة جملة تصيد الاخطاء كالتالي :

```
try
{
    ; code
}
catch (Exception e)
{
    ; code
}
```

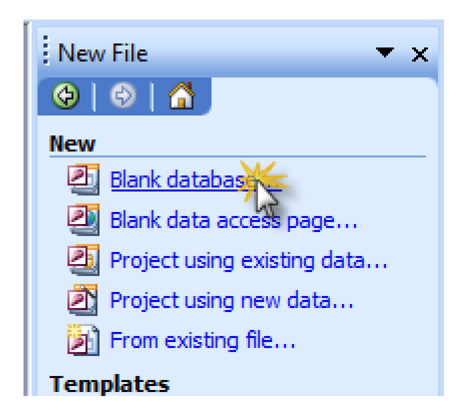
Exception :
يعتبر كنوع بيانات يقوم
بإدراج الخطأ في المتغير (e)

اولاً : نقوم بإنشاء قاعدة البيانات عن طريق برنامج الاكسس 2003 :

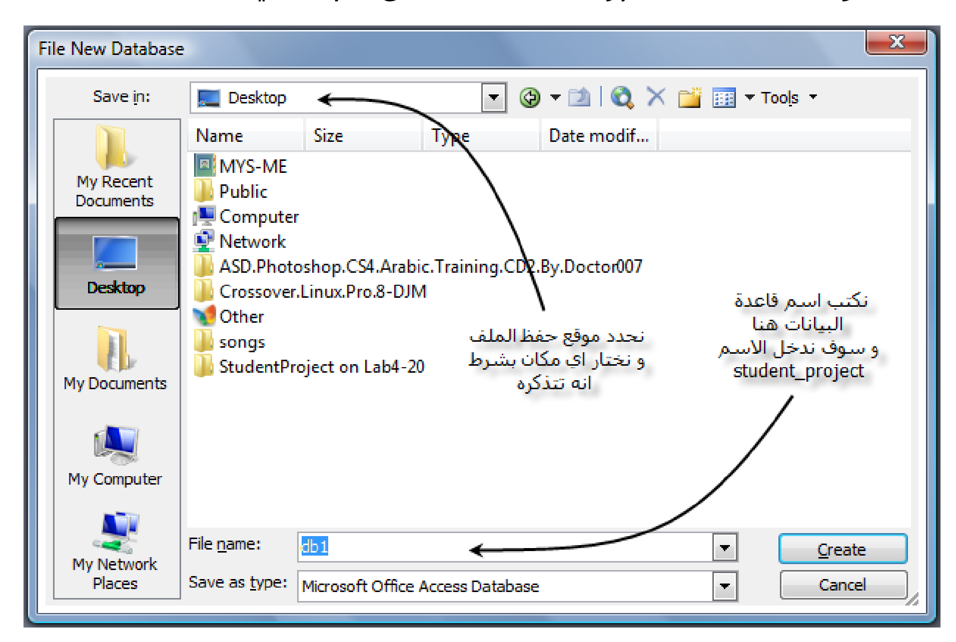
1. نقوم بالدخول الى برنامج الاكسس و نضغط على **جديد** كما في الصورة التالية :



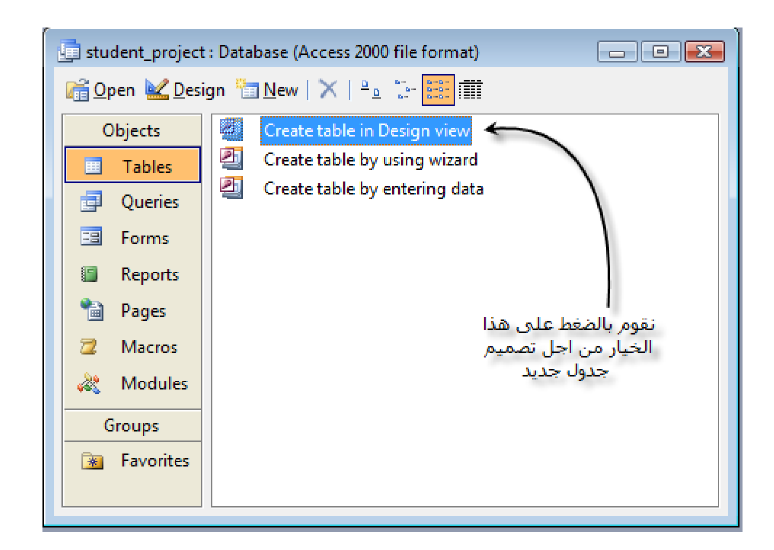
2. ثم نختار من النافذة الجديده التي ظهرت في الجانب **(Blank database)** :



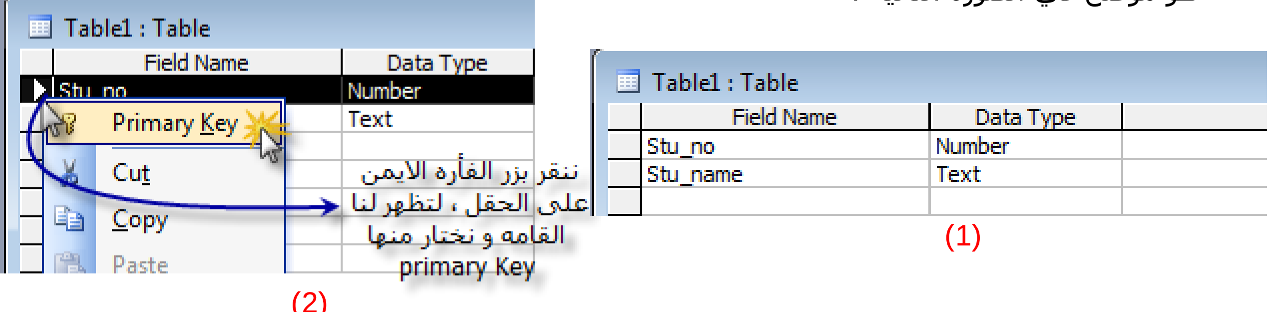
3. وبعد الخطوه السابقه ، ستظهر نافذه جديده ، نتبع فيها الاتي :



4. و سوف تظهر لنا نافذه جديده ، و نتبع الاتي :



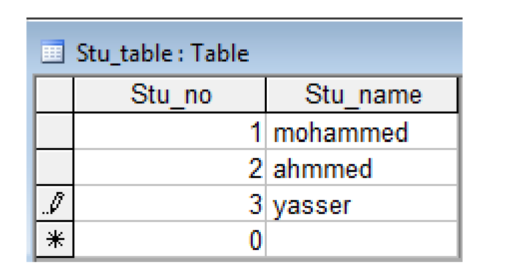
5. تظهر النافذه الجديده و نقوم بإنشاء الحقلين (رقم الطالب و اسم الطالب) و ندخل انواع الحقول مثل ما هو موضح بالصوره الاولى ، و لا ننسى بان نجعل رقم الطالب هو المفتاح الرئيسي كما هو موضح في الصورة الثانيه :



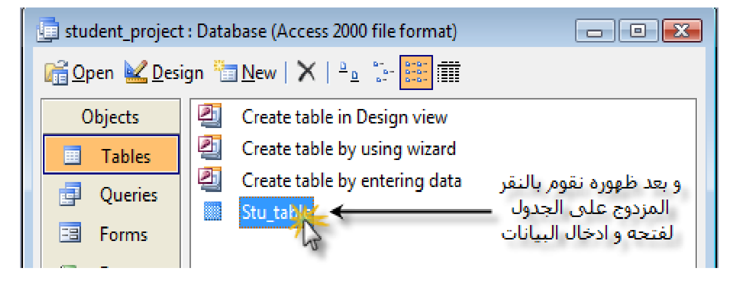
6. نقوم بإقفال النافذة السابقه الخاصه بتصميم الجدول ، وسوف تظهر لنا رساله تخبرنا بهل نريد حفظ الجدول ، اضغط **نعم** منها ، ثم تظهر نافذة اخرى تخبرنا بان نختار اسم للجدول ، و سوف ندخل فيها (**Stu_table**) .

7. و بعد حفظ الجدول سوف يظهر بهذا الشكل :

8. ندخل البيانات التي نريد ، و انا مثلاً ادخلت الاسماء التاليه :



	Stu_no	Stu_name
	1	mohammed
	2	ahmmed
✎	3	yasser
*	0	



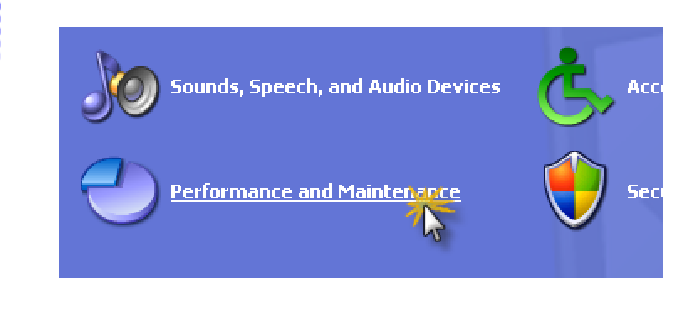
9. و بهذا نغلق نافذة ادخال البيانات للجدول ، و نغلق برنامج الاكسس ، ليصبح لدينا قاعدة بيانات من اجل عملية ربطها بمشروع مصمم بلغة جافا ، بالاضافه انها تحتوي على بيانات من اجل عرضها باستخدام المشروع .

ثانياً : بعد ان قمنا بإنشاء قاعدة البيانات ، نحتاج الى تفعيل الربط من اجل استدعائه من قبل المشروع الذي نريد تصميمه ، لذلك نتبع الخطوات التالية :

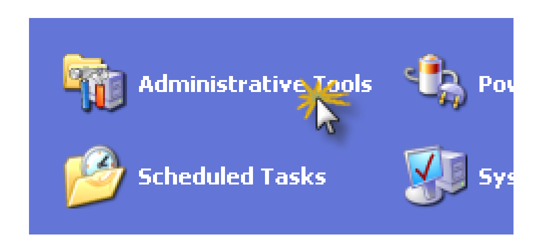
1. نقوم بفتح لوحة التحكم الموجوده في قائمة ابدأ .



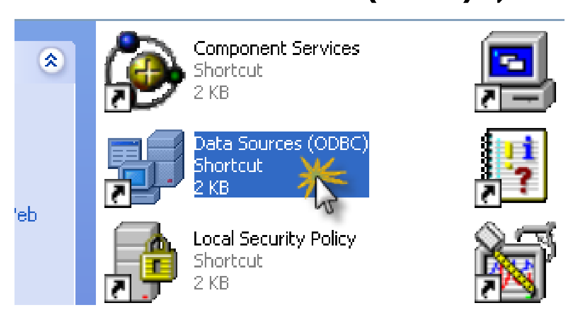
2. تظهر لنا نافذة جديده ، و نختار منها **performance and Maintenance** :



3. تظهر نافذة جديده و نختار منها **Administrative Tools** :

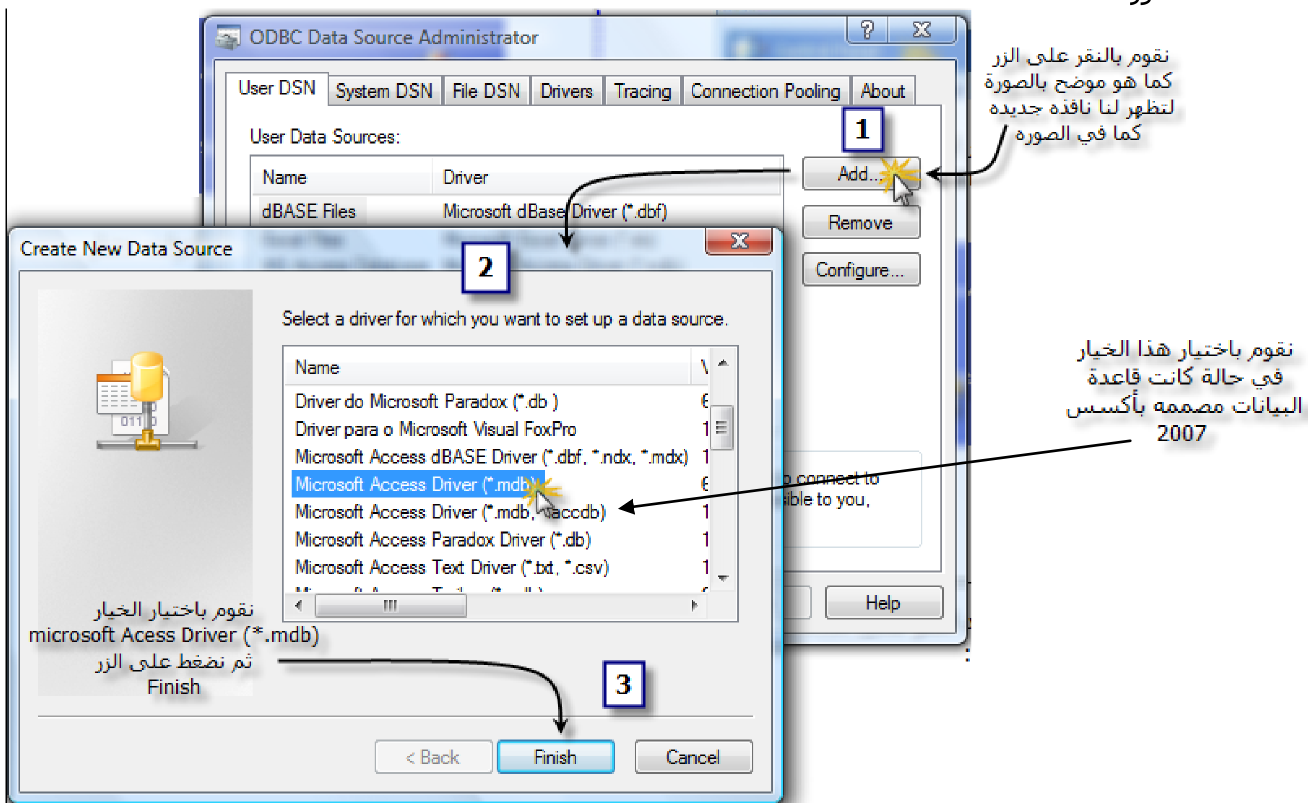


4. لتظهر لنا نافذة تحتوي على عدة ادوات، نختار منها **Data Sources (ODBC)** :

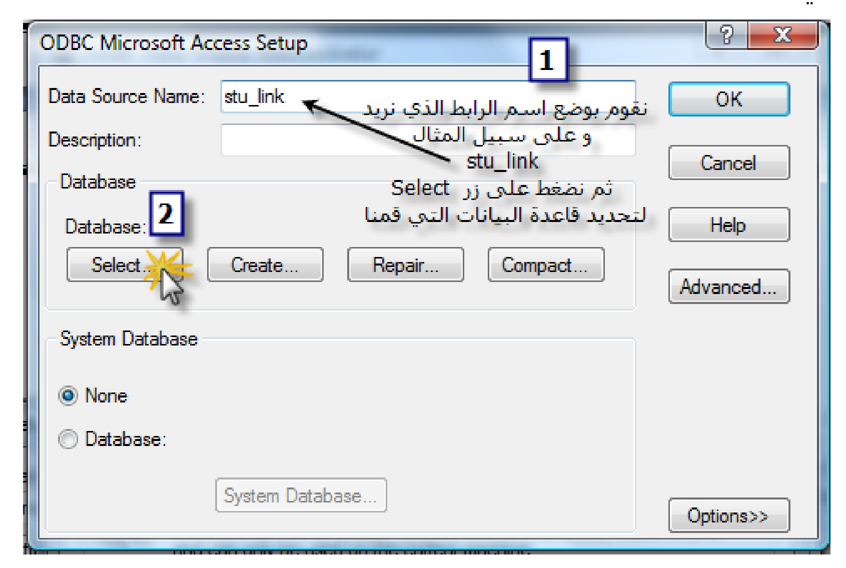


---

5. و بعد النقر على الاداة السابقة ، ستظهر نافذة جديده نضغط على الزر **add** كما هو موضح في الصورة :



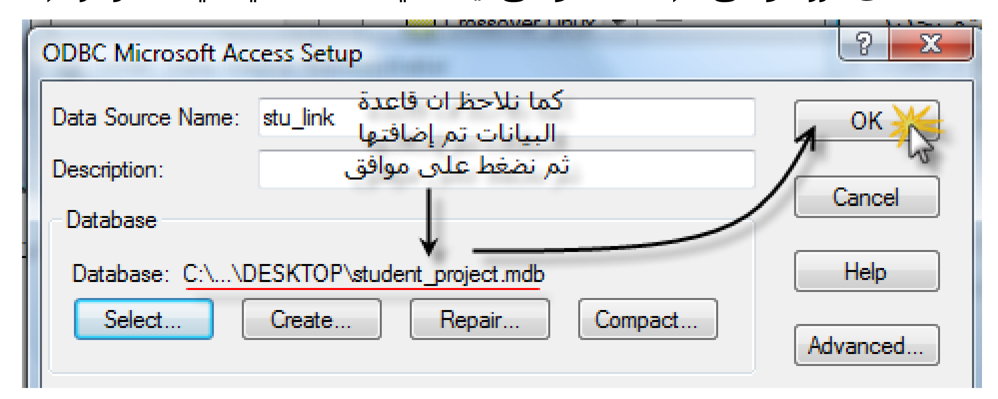
6. تظهر لنا نافذه اخرى جديده ، نقوم بكتابة اسم الربط الذي سنستخدمه في المشروع ، كما في الصورة التاليه :



7. ستظهر النافذة التي نقوم بتحديد قاعدة البيانات منها و نتبع الخطوات الموجوده على الصورة :



8. بعد الضغط على موافق سوف نعود الى النافذة الموجودة في الخطوة رقم سبعة ، ولكن هذه المره سوف نضغط على الزر **موافق** ثم نضغط **موافق** ايضاً في النافذه التي في الخطوه رقم خمسه :



◄ الان بقي فقط عملية ادراج الاكواد في المحرر ، و ربطه مع قاعدة البيانات عن طريق الشفرة البرمجيه :

← اولاً : نقوم استدعاء جميع الدوال التابعه للمكتبه SQL ، كما في السطر الاول بالكود التالي :

```
import java.sql.*;
import javax.swing.JOptionPane;
```

← اما بالنسبه للسطر الثاني فيمثل ادراج المكتبه التي تتعامل مع صناديق الحوار ، و يتم ادراج المكتبه بطريقتين إما يدوياً في بداية تصميم البرنامج او عن طريق كتابة شفرة احد صناديق الرسائل و سوف يقوم المحرر باضافة المكتبه مباشرة (اوتوماتيكيا) .

← و قبل ان نبدا بتصميم اي زر سوف نقوم بتعلم عملية ربط القاعدة بالبرنامج باستخدام الشفرات البرمجيه ، و من اجل التأكد من ان عملية الربط ناجحه سوف نقوم بجعل الاسم الاول الموجود في قاعدة البيانات يظهر لنا في محتوى رسالة حوار ، و الكود التالي يمثل عملية الربط وإظهار الاسم الاول في رساله ، و قد تم ترقيم اسطر الشفره من اجل شرح عمل كل سطر فيها :

```
public NewJFrame() {
    initComponents();

1   try
2   {
3    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
4    Connection con = DriverManager.getConnection("jdbc:odbc:stu_link");
5    Statement sql = con.createStatement();
6    ResultSet rs ;
7    rs = sql.executeQuery("Select * from stu_table");
8    rs.next();
9    JOptionPane.showMessageDialog(null,rs.getString("stu_name"));
10  }
11  catch(Exception e)
12  {
13   JOptionPane.showMessageDialog(null,e.getMessage());
14   System.exit(0);
    }
}
```

تمثل موقع كتابة الشفره البرمجيه و التي ستكون بداخل المشيد حتى يقوم بتشيد عملية الربط في بداية تشغيل البرنامج

السطر (1 ، 2 ، 10 ، 11) : و هي عبارة الكود الخاص بعملية تصيد الاخطاء (تم شرحها سابقاً) .

السطر (3) : يقوم هذا السطر بتحميل (JDBC Driver) ، الخاص بقواعد البيانات اكسس :

← Class : و هي عبارة عن مكتبه ، و يجب ان يكتب الحرف الاول منها كبير.

← forName : وهي عباره عن دالة تابعه للمكتبه Class لإستدعاء المشغل الخاص بعملية الربط .

السطر (4) : يقوم هذا السطر بعملية اتصال بقاعدة البيانات عن طريق الربط الذي قمنا بانشائه سابقاً و الذي قمنا بتسميته Stu_Link ، بحيث قمنا بتعريف متغير con من نوع اتصال الذي سوف يحوي عملية الربط ، بحيث نكتب jdbc:odbc كما هي و نقوم باضافة اسم الرابط الذي قمنا بانشائه سابقاً ليصبح jdbc:odbc:stu_link .

السطر (5) : تحدد جمل الاستعلام ، اي تحدد مدى امكانية تطبيق اوامر الاستعلام على قاعدة البيانات ، و سيتم شرح ذلك لاحقاً بالتفصيل - في صفحة (45) - .

السطر (6) : عرفنا متغير من نوع ResultSet بحيث سوف تخزن نتائج الاستعلام التي سوف نطبقها في السطر رقم (7) في داخل المتغير (rs) - و اعتبرها مثل المصفوفة اقل حجم لها هو صفر و اكبر حجم لها هو حجم الجدول نفسه - .

السطر (7) : تنفيذ جمل الاستعلام الخاصه بلغة SQL التي نريدها على قاعدة البيانات و اسناد نتائج التنفيذ في المتغير (rs) .

السطر (8) : بسبب ان المؤشر (الذي يقوم بالتأشير على سجلات الجداول) يكون خارج الجدول فنقوم بكتابة هذا السطر من اجل ان يؤشر على السجل الاول في الجدول .

السطر (9) : عباره عن رساله ، تقوم باظهار الاسم الموجود في السجل الاول و الذي سيكون (Mohammed) ، و الذي تم التاشير على سجله عن طريق الكود الموجود في السطر (8) .

ملاحظه : اذا اردنا ان نقوم باظهار الاسم الموجود في السجل الثاني نقوم بكتابة الكود الموجود في السطر رقم (8) مره اخرى ، و و بهذا ستظهر النتيجة في الرسالة (Ahmed) بدلاً من (Mohammed) ، و سيصبح الكود بهذا **الشكل** :

```
rs.next();
rs.next();
```

السطر (13) : يقوم هذا الكود بإظهار رساله تحتوي على الخطأ الذي حصل في البرنامج .

السطر (14) : عملية الخروج من الاجراء في حالة حدوث خطأ .

◄ و عند تنفيذ البرنامج سوف يتم اظهار رساله تحتوي على الاسم الاول في قاعدة البيانات ، كما في الشكل التالي :



ملاحظه :
و سيتم إظهار الرساله حتى قبل اظهار الفورم نفسه لان كود الرسالة مكتوب ضمن المشيد

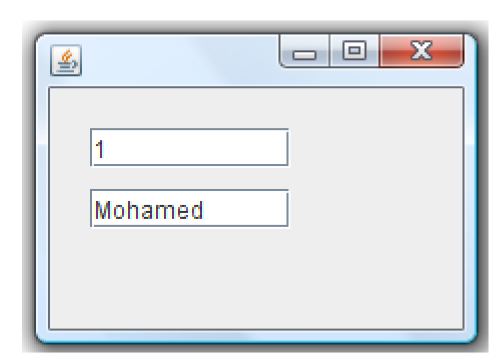
الى هنا عرفنا كيفية انشاء الربط بقاعدة البيانات ، و سوف نتعلم الان طريقة عرض رقم الطالب و اسم الطالب في حقلين من نوع jTextField بدلاً من إظهارها في رساله ، و سيتم ذلك عن طريق ادراج كائنات من نوع jTextField الى الفورم، و سيتم كتابة الشفره المناسبه و ذلك بالطريقة التالية:

1 ← نقوم بحذف الكود الموجود في السطر (9) ، و استبداله بالشفره التاليه :

```
jTextField1.setText(rs.getString("Stu_no"));
jTextField2.setText(rs.getString("Stu_name"));
```

عبارة عن دالة تقوم بتحويل القيمة الممررة اليها الى نصي حتى لو كان من نوع رقمي و كما نلاحظ ان الحقل الاول في الجدول يمثل ارقام الطلاب و من نوع رقمي

2 ← و عند تنفيذ الكود ستظهر النتيجه بهذا الشكل :



← بقي شي بسيط وهو معرفة الازرار التي تقوم بعملية التنقل بين السجلات في الجدول و عرضها في حقول jTextField :

اولاً : زر التنقل الى الامام و الذي بالشكل التالي [>] و الذي سيكون كوده بالشكل التالي :

```
private void btn_nextMouseClicked(java.awt.event.MouseEvent evt) {
    try
    {
        rs.next();
        jTextField1.setText(rs.getString("Stu_no"));
        jTextField2.setText(rs.getString("Stu_name"));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e.getMessage());
    }
}
```

كما تعلمنا سابقاً عند كتابة اي اوامر للتعامل مع قاعدة البيانات لا بد من كتابتها داخل جمل تصيد الاخطاء

← و عند كتابة الكود السابق تظهر خطوط حمراء متكسره ، بالرغم بان الكود صحيح ، وذلك لان المتغير الذي اسمه (rs) ليس متغير عام و إنما متغير خاص تم تعريفه داخل المشيد فقط لذلك نعود الى الكود الذي في المشيد و تحديداً السطر رقم (6) و حذفه و كتابته مره اخرى و لكن خارج المشيد ليصبح متغير عام ، كما في الشكل التالي :

```
public class NewJFrame extends javax.swing.JFrame {
    ResultSet rs ;
    /** Creates new form NewJFrame */
    public NewJFrame() {
        initComponents();

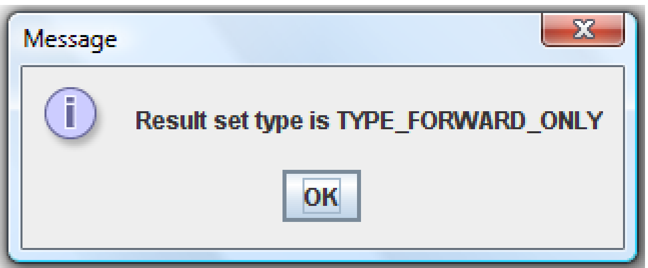
        try
        {
```

و بعد ذلك سوف نلاحظ إختفاء الخطوط الحمراء ، و سوف يتنفذ كود الزر بشكل سليم عند الضغط عليه .

ثانياً : زر التنقل الى الخلف و الذي بالشكل التالي [<] و الذي سيكون كوده بالشكل التالي :

```
private void btn_PrevMouseClicked(java.awt.event.MouseEvent evt) {
    try
    {
        rs.previous();
        jTextField1.setText(rs.getString("Stu_no"));
        jTextField2.setText(rs.getString("Stu_name"));
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e.getMessage());
    }
}
```

و لكن عند تنفيذ الكود ، و الضغط على الزر لن يتم يتنفيذ الكود و سوف تظهر رسالة تصيد الخطأ و بهذا المحتوى :



و معنى الرساله أن المغير الذي من نوع ResultSet بالشكل القياس ، (اي المؤشر يمشي للامام فقط) و يستخدم هذا النوع في عمليات اضافة الاسماء الى ليست او اضافتها الى جدول معين او نقوم بحساب عدد السجلات لان المؤشر يمشي للامام فقط ، و من اجل تعديل هذا الخيار نعود الى كود المشيد و بالتحديد السطر رقم (5) ، و يتم تعديل الكود بالشكل التالي :

```
Statement sql = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);
```

← و عند تنفيذ البرنامج مره اخرى ، سنرى ان الزر اصبح يعمل .

ثالثاً : زر التنقل الى اول سجل و الذي بالشكل التالي [<<] والذي سيكون كوده بالشكل التالي :

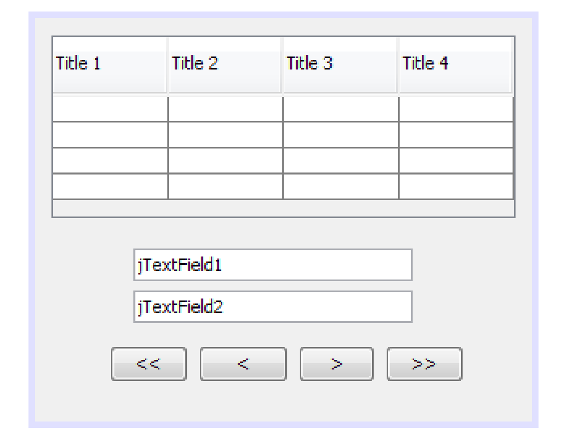
بنفس طريقة زر الرجوع الى الخلف و لكن بإبدال كلمة previous الى first ليصبح السطر هكذا :

```
rs.first();
```

رابعاً: زر التنقل الى اخر سجل و الذي بالشكل التالي [>>] والذي سيكون كوده بالشكل التالي :

بنفس طريقة زر الرجوع الى الخلف و لكن بإبدال كلمة previous الى last ليصبح السطر هكذا :

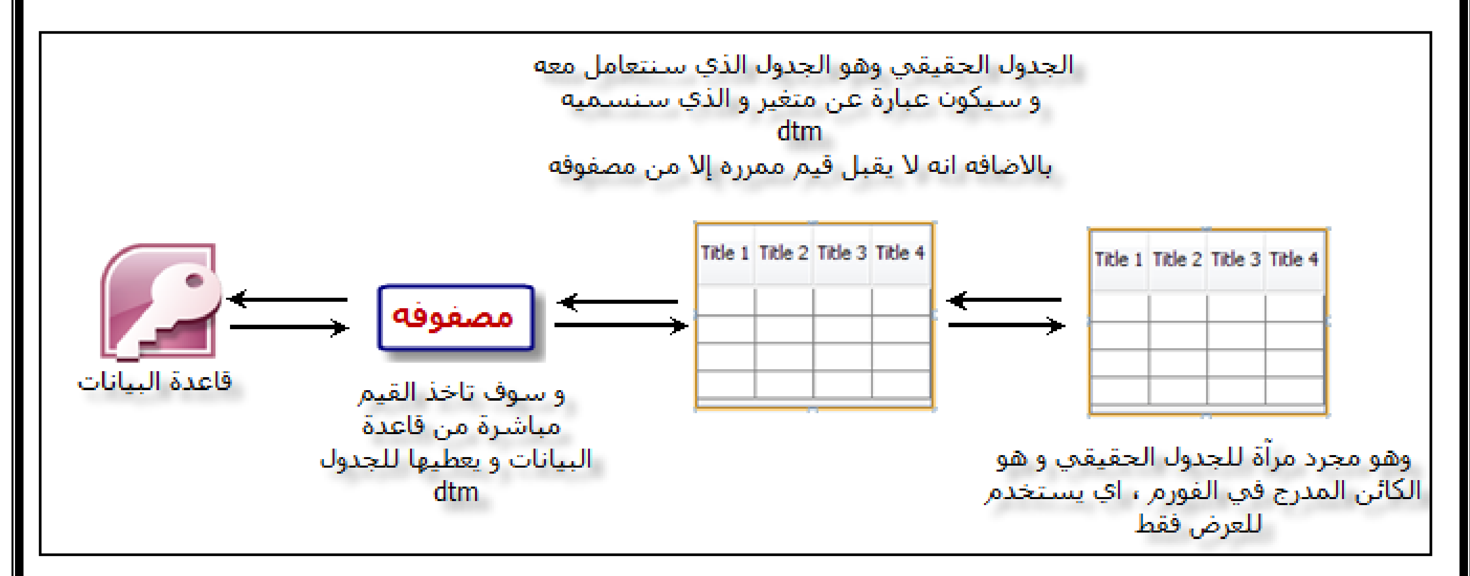
```
rs.last();
```



◄ الكائن jTable1 : وهو عبارة عن كائن على شكل جدول ، و يستخدم كأداة لعرض جداول قواعد البيانات، بالاضافه انه يمكن استخدامه مع المصفوفات لعرض محتوى المصفوفه .

← و سوف نقوم بإدراجه الى المشروع ليصبح الشكل النهائي للفورم كما في الشكل المقابل :

بقي الان كتابة الشفرة البرمجيه التي تقوم بإضافة محتوى قاعدة البيانات بداخل الجدول ، و لكن قبل ان ابدأ بعملية كتابة الشفرة ، سوف نتعرف على آلية الإضافه الى الجدول ، و ذلك بسبب ان الجدول لا يتعامل مباشرةً مع قواعد البيانات بل يحتاج الى وسيط و الشكل التالي يمثل طريقة الوسيط :



← و بعدما عرفنا طريقة الربط ، بقي أن نكتب الكود التالي بداخل المشيد و اعلى كود جملة تصيد الاخطاء **Try** ، كما في الشكل التالي :

قمنا بتعريف متغير يمثل جدول وسيط باسم dtm

هذا الكود يجعل الجدول المدرج على الفورم ياخذ بياناته من Model المسمى dtm

في هذه السطرين قمنا بتعريف عمودين في الجدول و اعطيناهما اسمين الاول رقم الطالب و الثاني اسم الطالب

```
/** Creates new form NewJFrame */
public NewJFrame() {
    initComponents();

    DefaultTableModel dtm = new DefaultTableModel();
    jTable1.setModel(dtm);
    dtm.addColumn("Student no");
    dtm.addColumn("Student name");

    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

➔ بقي فقط عملية ملئ الجدول بالبيانات لذلك نقوم باضافة حلقة التكرار while بداخل كود try الموجود بداخل المشيد :

```
try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection con = DriverManager.getConnection("jdbc:odbc:Stu_Link");
    Statement sql = con.createStatement(ResultSet.TYPE_SCROLL_INSEN
    rs = sql.executeQuery("Select * from Stu_table");

     while(rs.next())
    {
        Object ary[] = {rs.getInt("Stu_no"),rs.getString("Stu_name")};
        dtm.addRow(ary);
    }

    rs.first();
    jTextField1.setText(rs.getString("Stu_no"));
    jTextField2.setText(rs.getString("Stu_name"));
}
catch(Exception e)
{
```

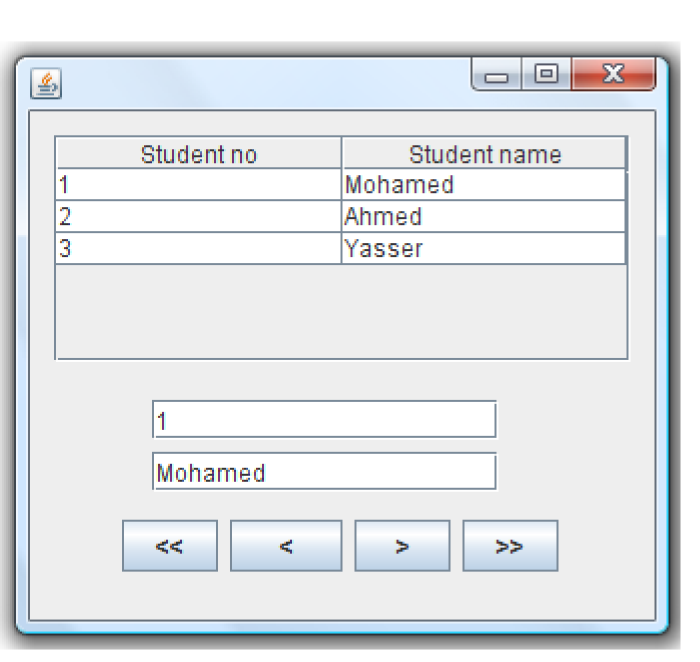
هذا السطر يعني ، قم بعملية الدوران طالما ما زال المؤشر ينتقل الى السجل التالي

مصفوفه من نوع Object و ذلك لان المصفوفه الرقميه لا تقبل نص و العكس صحيح ، لان الجدول الموجود في قاعدة البيانات يحتوي على حقل رقمي و حقل نصي لذلك قمنا بتعريف متغير يجمع جميع الانواع

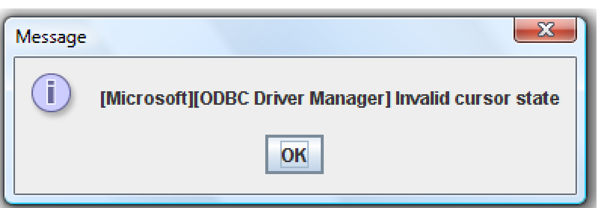
عملية إضافة صف (سجل) جديد في الجدول dtm ، بمحتوى القيم الموجودة بداخل المصفوفة

قمنا بجعل المؤشر يعود الى اول سجل بسبب انه اصبح الان في اخر سجل بسبب الدوارة ، ولكي تكون عملية التنقل بالازرار من اول سجل في الجدول

و عند تنفيذ البرنامج سيظهر بالشكل التالي :



مشكله و حلها : عندما نقوم بالضغط على زر التالي و نصل الى اخر سجل و نقوم بالضغط مره اخيره على الزر تظهر رسالة خطأ بانه لا يوجد سجل تالي ، كما في الشكل التالي :



ولحل ذلك نقوم بتعديل كود زر التالي بالشكل التالي حتى لا تظهر هذه الرساله مره اخرى:

```
private void btn_nextMouseClicked(java.awt.event.MouseEvent evt) {
    try
    {
        if(rs.isLast())
        {
            btn_next.setEnabled(false);
        }
        else
        {
            rs.next();
            jTextField1.setText(rs.getString("Stu_no"));
            jTextField2.setText(rs.getString("Stu_name"));
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e.getMessage());
    }
}
```

و تعني عند وصول المؤشر الى السجل الاخير قم بإلغاء تفعيل زر التالي

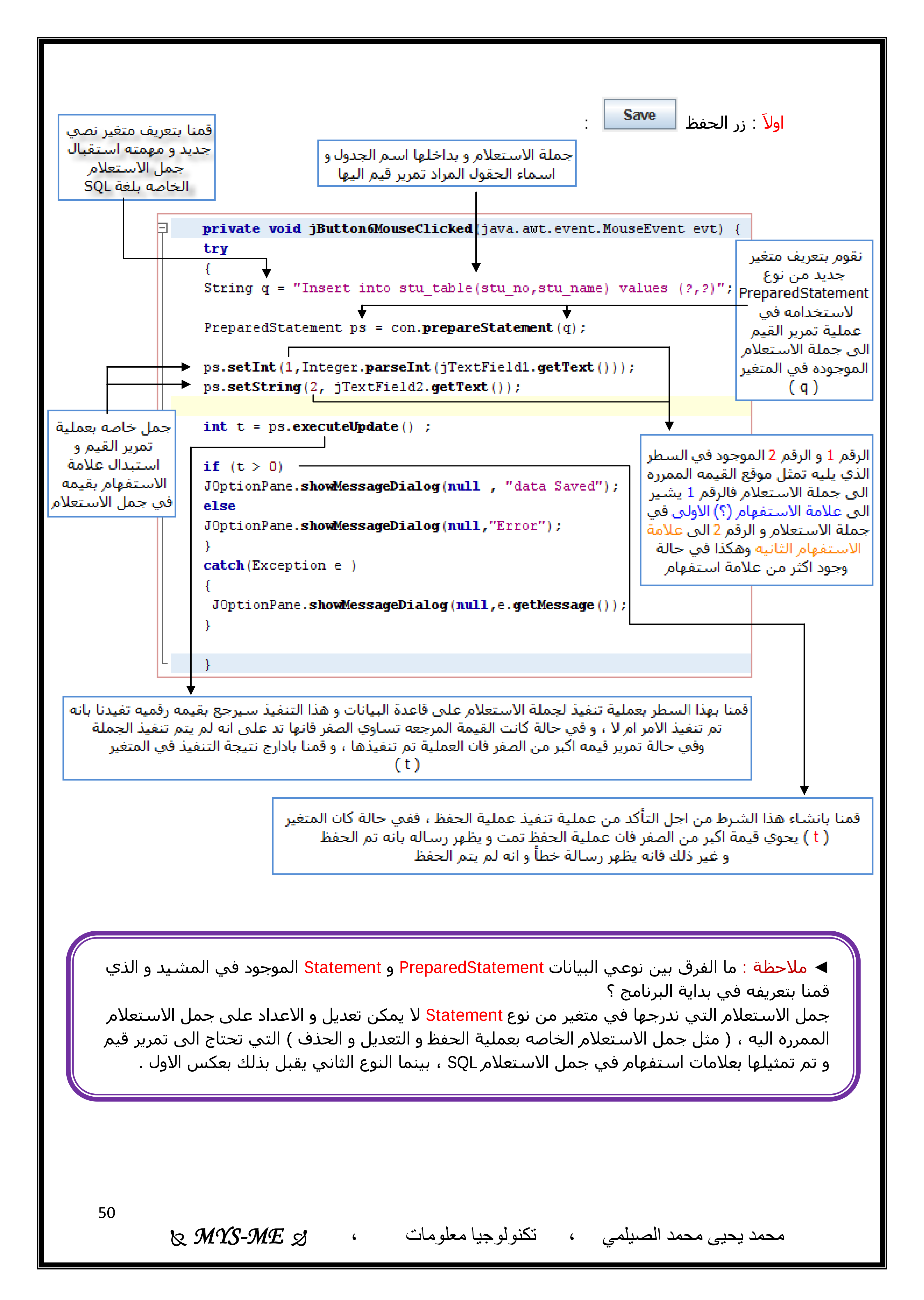
ملاحظة : يمكن عمل هذا الكود مع الزر السابق ايضاً ، مع تغير ما يلزم لتلافي رسائل التحذير .

* بعدما عرفنا ازرار التنقل بقي لنا ان نتعلم ازرار : الحفظ ، و الحذف ، و التعديل .
و لكن قبل ان نبدأ بكتابة الاكواد الخاصه بالحفظ و غيرها من الازرار ، فإننا سوف نحتاج الى المتغيرات التي قمنا بتعريفها في المشيد، مثل المتغير الخاص بالاتصال (con) وسوف نقوم بمسحها من المشيد و اعادة تعريفها بشكل عام من اجل استخدامها من اي مكان بالبرنامج ، بهذا الشكل :

```
public class NewJFrame extends javax.swing.JFrame {
    ResultSet rs ;
    Statement sql ;
    Connection con ;

    /** Creates new form NewJFrame */
    public NewJFrame() {
        initComponents();
```

هذه هي المتغيرات وكما نلاحظ اصبحت خارج المشيد



اولآ : زر الحفظ Save :

قمنا بتعريف متغير نصي جديد و مهمته استقبال جمل الاستعلام الخاصة بلغة SQL

جملة الاستعلام و بداخلها اسم الجدول و اسماء الحقول المراد تمرير قيم اليها

```
private void jButton6MouseClicked(java.awt.event.MouseEvent evt) {
try
{
String q = "Insert into stu_table(stu_no,stu_name) values (?,?)";

PreparedStatement ps = con.prepareStatement(q);

ps.setInt(1,Integer.parseInt(jTextField1.getText()));
ps.setString(2, jTextField2.getText());

int t = ps.executeUpdate() ;

if (t > 0)
JOptionPane.showMessageDialog(null , "data Saved");
else
JOptionPane.showMessageDialog(null,"Error");
}
catch(Exception e )
{
 JOptionPane.showMessageDialog(null,e.getMessage());
}

}
```

نقوم بتعريف متغير جديد من نوع PreparedStatement لاستخدامه في عملية تمرير القيم الى جملة الاستعلام الموجودة في المتغير (q)

جمل خاصة بعملية تمرير القيم و استبدال علامة الاستفهام بقيمه في جمل الاستعلام

الرقم 1 و الرقم 2 الموجود في السطر الذي يليه تمثل موقع القيمه الممررة الى جملة الاستعلام فالرقم 1 يشير الى علامة الاستفهام (؟) الاولى في جملة الاستعلام و الرقم 2 الى علامة الاستفهام الثانية وهكذا في حالة وجود اكثر من علامة استفهام

قمنا بهذا السطر بعملية تنفيذ لجملة الاستعلام على قاعدة البيانات و هذا التنفيذ سيرجع بقيمه رقميه تفيدنا بانه تم تنفيذ الامر ام لا ، و في حالة كانت القيمة المرجعه تساوي الصفر فانها تد على انه لم يتم تنفيذ الجملة وفي حالة تمرير قيمه اكبر من الصفر فان العملية تم تنفيذها ، و قمنا بادراج نتيجة التنفيذ في المتغير (t)

قمنا بانشاء هذا الشرط من اجل التأكد من عملية تنفيذ عملية الحفظ ، ففي حالة كان المتغير (t) يحوي قيمة اكبر من الصفر فان عملية الحفظ تمت و يظهر رسالة بانه تم الحفظ و غير ذلك فانه يظهر رسالة خطأ و انه لم يتم الحفظ

◄ ملاحظة : ما الفرق بين نوعي البيانات PreparedStatement و Statement الموجود في المشيد و الذي قمنا بتعريفه في بداية البرنامج ؟

جمل الاستعلام التي ندرجها في متغير من نوع Statement لا يمكن تعديل و الاعداد على جمل الاستعلام الممرره اليه ، (مثل جمل الاستعلام الخاصة بعملية الحفظ و التعديل و الحذف) التي تحتاج الى تمرير قيم و تم تمثيلها بعلامات استفهام في جمل الاستعلام SQL ، بينما النوع الثاني يقبل بذلك بعكس الاول .

ثانياً : زر التعديل update :

بنفس كود و نفس الاسلوب الذي قمنا باتباعه مع زر الحفظ ولكن الاختلاف فقط بعملية تمرير القيم و جملة الاستعلام الخاصه بلغة SQL ، و الكود التالي هو كود التعديل :

```
private void jButton7MouseClicked(java.awt.event.MouseEvent evt) {
try
    {
    String q = "update stu_table set stu_name = ? " +
            "where stu_no = ? ";

    PreparedStatement ps = con.prepareStatement(q);

    ps.setString(1, jTextField2.getText());
    ps.setInt(2,Integer.parseInt(jTextField1.getText()));

    int t = ps.executeUpdate() ;

    if (t > 0)
    JOptionPane.showMessageDialog(null , "data updated");
    else
    JOptionPane.showMessageDialog(null,"Error");

    }
    catch(Exception e )
    {
     JOptionPane.showMessageDialog(null,e.getMessage());
    }
    }
```

جملة الاستعلام الخاصة بعملية التعديل ، وعملها في هذا الكود تعديل الاسم الذي نريد بدلالة رقم الاسم الذي سندخله ، فمثلاً نريد تعديل الاسم محمد الذي رقمه (1) ، فإننا نمرر لجملة الاستعلام الرقم الخاص بالاسم محمد و الذي يمثل الرقم (1) ، و نكتب الاسم الجديد لمحمد و ستقوم جملة الاستعلام بالبحث عن الرقم (1) في الجدول واضافة الاسم المعدل مثلاً (محمد222) .

قمنا في هذا الكود بقلب الارقام ، لاننا نحتاج الى تمرير رقم الطالب في الاستفام الثانيه في جملة الاستعلام الموجوده في السطر الاول لذلك نكتب (2) لان ترتيب علامة الاستفهام في الجمله هو الثاني . و نحتاج الى تمرير اسم الطالب في علامة الاستفهام الاولى لذلك نكتب (1) .

◄ ملاحظة : السطر الاول في الكود السابق و المكتوب في بهذه الطريقة :

```
String q = "update stu_table set stu_name = ? " +
        "where stu_no = ? ";
```

هو نفس السطر التالي :

```
String q = "update stu_table set stu_name = ? where stu_no = ? ";
```

و قد قمنا بكتابته في شكله الاول من اجل تقصير سطر الكود البرمجي ، بحيث قسم جملة الاستعلام الى جملتين و قام البرنامج بوضع علامة دمج النصوص (+) تمثل ان هذه الجملتين المنفصلتين بالسطور ولكنها جملة واحد بالاصل ، و يتم تنفيذ هذه الطريقه بوضع مؤشر الفأرة على المنتصف و ضغط زر Enter من على لوحة المفاتيح للنزول الى السطر التالي و سيقوم المحرر تلقائياً بكتابة اشارات دمج النصوص .

ثالثاً : زر الحذف delete :

بنفس الاسلوب و بنفس الكود مع تغير جملة الاستعلام فقط و القيم الممرره و في جملة الاستعلام هذه سوف نحتاج الى قيمه واحده ، و تمثل تمرير رقم الاسم الذي نريد حذفه ، و سيكون الكود بالشكل التالي :

```
private void jButton8MouseClicked(java.awt.event.MouseEvent evt) {
    try
    {
    String q = "DELETE FROM stu_table WHERE stu_no = ? ";
    PreparedStatement ps = con.prepareStatement(q);

    ps.setInt(1,Integer.parseInt(jTextField1.getText()));

    int t = ps.executeUpdate() ;

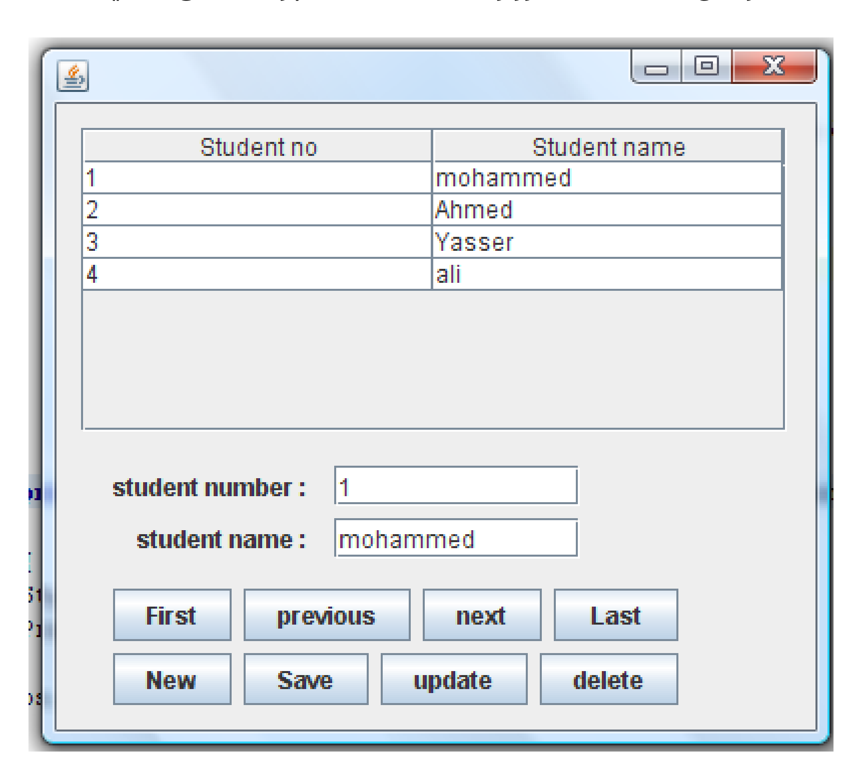
    if (t > 0)
    JOptionPane.showMessageDialog(null , "data Deleted");
    else
    JOptionPane.showMessageDialog(null,"Error");
    }
    catch(Exception e )
    {
     JOptionPane.showMessageDialog(null,e.getMessage());
    }
    }
```

رابعاً : زر جديد New :

و يقوم هذا الزر بعملية تفريغ محتوى الحقول من اجل تسجيل الرقم و الاسم الجديدان ثم الضغط على زر الحفظ من اجل حفظها .

```
private void jButton5MouseClicked(java.awt.event.MouseEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    }
```

➔ و عند تنفيذ البرنامج بعد اضافة الازرار الجديده ، سيظهر بالشكل الاتي :



✱ الى هنا انتهينا من هذا الكتاب ، مع تمنياتي للجميع بالتوفيق .



- تم و بحمد الله -

MYS-ME